NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

MBA PROFESSIONAL REPORT

Heuristics for Solving Problem of Evacuating Non-Ambulatory People in a Short-Notice Disaster

---

By: Andres Pico, and
Yong Hui Ronny Tan
December 2012

Advisors: Aruna Apte,
Susan Heath

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** December 2012	**3. REPORT TYPE AND DATES COVERED** MBA Professional Report

4. TITLE AND SUBTITLE Heuristics for Solving Problem of Evacuating Non-Ambulatory People in a Short-Notice Disaster	**5. FUNDING NUMBERS**
6. AUTHOR(S) Andres Pico, and Yong Hui Ronny Tan	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ______N/A________.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

As Humanitarian Assistance and Disaster Relief (HADR) operations gain importance, a number of problems become evident. The time-sensitive problem of evacuating non-ambulatory people from a disaster area proves to be a challenging combinatorial optimization problem. The scope of the problem is defined by drawing analogies to similar vehicle routing problems that have been previously addressed. Based on the basic Max-Min Ant System (MMAS) algorithm modeled after the behavior of ants seeking food, potential solution approaches to this problem are enhanced to improve quality and efficiency by hybridizing features such as a best solution list, elite ants, ranked contribution system, and heuristic procedures during route construction. Using a Nearly-Orthogonal Latin Hypercubes (NOLH) experimental design, the algorithm parameters are tuned for best empirical performance for a range of test scenarios.

14. SUBJECT TERMS Disaster evacuation, Vehicle Routing Problem	**15. NUMBER OF PAGES** 161
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89)

Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

HEURISTICS FOR SOLVING PROBLEM OF EVACUATING NON-AMBULATORY PEOPLE IN A SHORT-NOTICE DISASTER

Andres Pico, Lieutenant Commander, United States Navy
Yong Hui Ronny Tan, Ministry of Defense, Singapore

Submitted in partial fulfillment of the requirements for the degree of

MASTER OF BUSINESS ADMINISTRATION

from the

NAVAL POSTGRADUATE SCHOOL
December 2012

Authors: ____
Andres Pico

____
Yong Hui Ronny Tan

Approved by: ____
Aruna Apte, Co-Advisor

____
Susan Heath, Co-Advisor

____
William R. Gates, Dean
Graduate School of Business and Public Policy

THIS PAGE INTENTIONALLY LEFT BLANK

HEURISTICS FOR SOLVING PROBLEM OF EVACUATING NON-AMBULATORY PEOPLE IN A SHORT-NOTICE DISASTER

ABSTRACT

As Humanitarian Assistance and Disaster Relief (HADR) operations gain importance, a number of problems become evident. The time-sensitive problem of evacuating non-ambulatory people from a disaster area proves to be a challenging combinatorial optimization problem. The scope of the problem is defined by drawing analogies to similar vehicle routing problems that have been previously addressed. Based on the basic Max-Min Ant System (MMAS) algorithm modeled after the behavior of ants seeking food, potential solution approaches to this problem are enhanced to improve quality and efficiency by hybridizing features such as a best solution list, elite ants, ranked contribution system, and heuristic procedures during route construction. Using a Nearly-Orthogonal Latin Hypercubes (NOLH) experimental design, the algorithm parameters are tuned for best empirical performance for a range of test scenarios.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ACO	Ant Colony Optimization
AS	Ant System
AVRP	Asymmetric Vehicle Routing Problem
B&B	Branch and Bound
B&C	Branch and Cut
BPP	Bin Packing Problem
CCRP	Capacity Constrained Route Planner
CO	Combinatorial Optimzation
CRED	Center for Research on the Epidemiology of the Disaster
CVRP	Constrained Vehicle Routing Problem
DCVRP	Distance Constrained Vehicle Routing Problem
DE	Differential Evolution
EA	Evolutionary Algorithm
FEMA	Federal Emergency Management Agency
FLGA	Fuzzy Logic Guided Genetic Algorithm
FLP	Facility Location Problem
FSMVRP	Fleet Size and Mix Vehicle Routing Problem
GA	Genetic Algorithm
GAP	Generalized Assignment Problem
GRASP	Greedy Randomized Adaptive Search Procedure
HAS	Hybrid Ant System
HVRP	Heterogeneous Vehicle Routing Problem
IGA	Improved Genetic Algorithm
IFAS	Institute of Food and Agricultural Sciences
MACS	Multiple Ant Colony Systems
MMAS	Max-Min Ant System
NOLH	Nearly-Orthogonal Latin Hypercubes
NP	Non-deterministic Polynomial-time
OB-VRP	Overburdened Vehicle Routing Problem
PCTSP	Prize Collected Traveling Salesman Problem
QAP	Quadratic Assignment Problem
RCL	Restricted Candidate List
SA	Simulated Annealing
SCB	Set-Covering-Based
SCP	Set Covering Problem
SO	Strategic Oscillation
SPP	Set-Partitioning Problem
STSP	Symmetric Traveling Salesman Problem
TS	Tabu Search
TSP	Traveling Salesman Problem
TSPPD	Traveling Salesman Problem with Pickup and Delivery

TSPTW	Traveling Salesman Problem with Time Windows
VRPB	Vehicle Routing Problem with Backhauls
VRPBTW	Vehicle Routing Problem with Backhauls and Time Windows
VRPMPD	Vehicle Routing Problem with Mixed Pickup and Delivery
VRPPD	Vehicle Routing Problem with Pickup and Delivery
VRPPDTW	Vehicle Routing Problem with Pickup and Delivery and Time Windows
VRPSPD	Vehicle Routing Problems with Simultaneous Pickup and Delivery
VRPTW	Vehicle Routing Problems with Time Windows

ACKNOWLEDGMENTS

The authors would like to thank Aruna and Susan for providing invaluable guidance and advice during this journey of discovery into the world of humanitarian logistics.

Ronny would like to deeply thank Ling-Er for being his loving and dedicated wife. She has offered endless sacrificial support in taking care of the babies while he was constantly in school. He also wants to the babies to know that no matter how much exasperation they bestow upon him, he will always love them as their father.

Andres would like to thank his wife Shannon for her incredible patience while he spent hours on the computer babbling about ants and their pheromones. Special thanks also goes to his toddler sons Andy and Alex, who provided unceasing entertainment during the entire process, and some thanks goes to the North Company whose Gun Muffler ear muffs capably regulated the amount of said toddler entertainment to barely audible levels.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

According to the Post-Katrina Emergency Management Reform Act of 2006, state and local governments have the responsibility to coordinate evacuation plans for all populations, including those with disabilities (Apte & Heath, 2011; Congressional Research Service, 2005). Yet, recent natural disasters such as the 2004 Indian Ocean tsunami, 2005 Hurricane Katrina in the United States, 2010 Haitian and 2011 Turkey earthquakes, and 2011 Great Eastern Coast of Japan tsunami have exposed the shortcomings in humanitarian logistics planning for disaster, especially for the critical evacuation and response stage.

A. THESIS PROBLEM

This study focuses on the problem of assisted evacuation in a short-notice disaster. This is distinct from the self-evacuation problem where the concern is with how individuals can maximize their survival chances by retreating from the disaster area on their own capability, e.g., on foot or self-driven vehicles. Research interests in this latter domain pertain more to optimizing pedestrian and vehicle traffic flow, for example, by designing shortest possible exit routes, or manipulating traffic directions and intersection stopping times. In contrast, assisted evacuation is concerned with how government authorities can utilize their facilities, manpower and other resources to provide assistance to citizens who cannot self-evacuate, primarily due to lack of private transportation means or disability. Typically, an assisted evacuation plan requires such people to assemble at selected central locations to board vehicles in order to be mass-evacuated. Unfortunately, this type of plan may not be amenable to those who are unable to move themselves to the designated assembly locations. At the same time, local authorities face many constraints such as limited number and variety of evacuation vehicles, diverse mobility level of evacuees, available time, etc. Key to minimizing loss of life often relies on quick and optimal determination of vehicle assignment and routes.

The thesis problem, to be formally defined in Chapter III, aims to develop a routing model that sends vehicles to pick up and evacuate as many people as possible from their homes to a common shelter, within given constraints. The model needs to generate routes assigning people to vehicles in an optimal sequence, while accommodating the various levels of disability and complete the evacuation within a limited time window, taking into consideration varied loading and unloading times. It is assumed that there is a known list of assisted evacuees, together with their locations and disability level mapped to the type of vehicle required (people with lower severity can be transported on vehicles designed for people with higher severity but not vice versa). There is no prioritization of people, and in the situation where there are several people at a single location, it is not assumed that a vehicle has to pick up all of them simultaneously.

While the practitioner realm within which disaster evacuation falls is the relatively new field of humanitarian logistics research, the academic discipline underpinning such general disaster evacuation problems is the field of Combinatorial Optimization (CO), and, more specifically, an important class of problems known as Vehicle Routing Problems (VRPs). The VRP primarily deals with the distribution and transportation of people and commodities. The problem can be generally described as the determination of an optimal set of routes for a fleet of vehicles to serve a given set of customer needs (Toth & Vigo, 2002). Numerous variants have extended the basic VRP model in order to address real-world problems that are often more complicated and dynamic in nature, e.g., restricting the capacity of the vehicle(s) and specifying fixed service times for visiting customers. The variant within the VRP family that is relevant to the broad disaster evacuation problem is the VRP with Pickup and Delivery (VRPPD). The related subvariant of interest for this study is the VRP with Mixed Pickup and Delivery (VRPMPD). However, unlike the traditional VRPMPD, the thesis problem is saddled with additional complicating factors and constraints such as:

- Multiplicity and heterogeneity of vehicles and originating depots;
- Ranked heterogeneity of customers' transportation need levels which must correspond to vehicle capabilities;

- Multiplicity of tours the vehicles can make;
- Total allotted time available; and
- An objective function that focuses on penalizing number of un-served customers, rather than distance or vehicle cost.

To highlight the complicating difference between the thesis problem and prior VRP's, it will henceforth be called the Overburdened VRP (OB-VRP).

B. THESIS CONTRIBUTION AND RESEARCH OBJECTIVES

The primary contributions of the thesis are three-fold. Firstly, it tackles the complex OB-VRP first identified and formulated in Apte and Heath (2011), for which, to the authors' knowledge, a solution has yet to appear in literature. The proposed solution determines the evacuation routes, vehicle loads, and vehicle route-tour schedule (when to evacuate with which vehicle carrying how much load via which route on which tour). The thesis also applies an objective function that is based on number of un-served customers versus the more conventional time/distance travelling cost and/or vehicle cost found in literature. The implication of such a choice is that to improve the cost function, an un-served customer must be added to one of the routes, thereby increasing time, contrary to traditional VRPs.

Secondly, the thesis proposes a solution approach that offers feasibility, proximity to optimality, scalable speed and implementation elegance. This is key, given that despite advances in algorithmic approaches, solving the VRP to optimality remains elusive for very large problem sizes, with limitations on exact methods, and difficulty in directly applying approximation algorithms for the basic VRP and its classic variants onto the significantly more complex OB-VRP. Further, due to the many problem constraints that often apply, obtaining a feasible solution is often a challenge in itself, with repair workarounds inevitably rendering the solution algorithm less elegant and slowing down the optimization process.

Lastly, the thesis demonstrates the use of an efficient space-filling experimental design method based on Nearly-Orthogonal Latin Hypercubes (NOLH) to determine the best parameters for the proposed solution algorithm to cater for a broad range of test scenarios as first demonstrated in Heath, Bard and Morrice (2012).

C. ORGANIZATION OF THESIS

Chapter II reviews the literature from both thematic and methodological perspectives. The thematic analysis sets up the backdrop by looking at humanitarian logistics and disaster response research in general. The methodological survey examines the VRP family and discusses general solutions approaches, including exact, heuristic and meta-heuristic approaches. Chapter III describes and formally defines the OB-VRP as a graph theoretic model, before providing a linear mixed integer formulation first presented in Apte and Heath (2011). Chapter IV lays out the proposed solution approach. It includes a discussion of preliminary studies and the insights gained in its development, as well as a detailed exposition of the complete solution algorithm. Chapter V documents the numerical computational results of the solution approach using stylized data. Chapter VI summarizes the work undertaken and offers suggestions for future research directions.

II. LITERATURE REVIEW

This chapter provides the background from both thematic and methodological perspectives. The thematic review discusses humanitarian logistics and disaster response research in general, as well as reviews the literature vis-à-vis the evacuation planning problem. The methodological survey explores the VRP family, with emphasis on the related subvariants of the Vehicle Routing Problem with Pickups and Deliveries (VRPPD). General solution approaches to CO problems and VRPs are discussed. Brief descriptions and comparison of exact solution, heuristic and meta-heuristic approaches are provided.

A. HUMANITARIAN LOGISTICS: A THEMATIC REVIEW

This section looks at disaster and humanitarian aid trends, explains how humanitarian logistics is modeled as a supply chain and gives an overview of the evacuation planning problem

1. Disaster and Humanitarian Aid Trends

A considerable number of the world's population has suffered in recent years as a result of the increasing frequency and magnitude of disasters (Figure 1), with a disaster being defined by the U.S. Federal Emergency Management Agency (FEMA) as an event that causes 100 deaths or 100 human injuries or damage worth U.S.$ 1 million. The Center for Research on the Epidemiology of the Disaster (CRED) reports that in 2010 alone, 385 natural disasters killed 297,000 people worldwide, affecting over 217 million more and causing U.S.$ 123.9 billion of economic damages. In particular, short-notice disasters such as floods (e.g., May–August 2010 flood in People's Republic of China and October–December 2010 flood in Thailand) make up four out of the top five natural disasters in terms of number of victims affected, while accounting for 92% of the victims in the top ten disasters (Guha-Sapir, Vos, Below & Ponserre, 2011). Thomas and Kopczak (2005) forecast natural and man-made disasters over the next 50 years to increase five-fold.

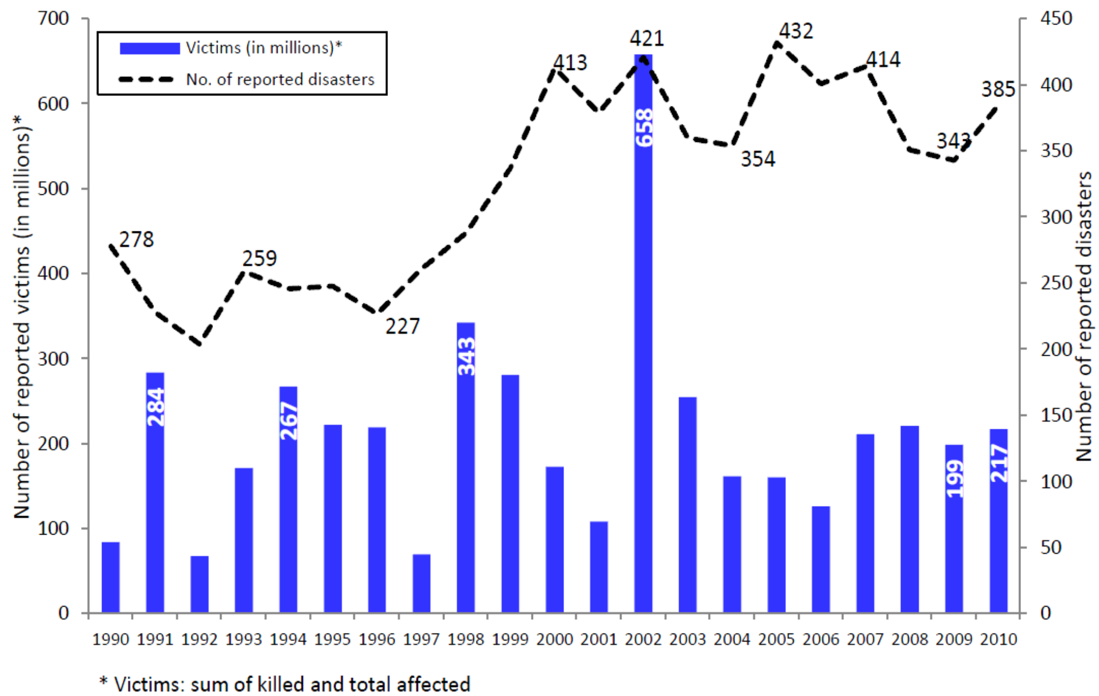


Figure 1. Trends in natural disaster incidence and victims (From Guha-Sapir et al., 2011)

With the 2004 budgets of the top 10 humanitarian agencies exceeding $14 billion in total, the logistics of aid has attracted increasing scrutiny (Thomas & Kopczak, 2005). Yet, recent humanitarian responses to the 2010 Haitian and 2011 Turkey earthquakes, the 2005 Hurricane Katrina in the United States, and the 2004 Indian Ocean and 2011 Great Eastern Coast of Japan tsunami have largely been neither effective nor efficient (Apte, 2009). Causes of these inefficiencies are many, including the sheer size and scope of such disasters, but with rising scrutiny, reports of how public officials are ill-prepared and fail to mitigate the resulting damage and loss of lives has become plentiful (Apte, 2009). For instance, during Hurricane Katrina, “beginning with the evacuation orders before the hurricane landfall, some public officials did not know what those right steps might be”

(Nieburg, Waldman, & Krumm, 2005). The Katrina evacuation fiasco thus points to the need for more effective mass evacuation planning.

In particular, McGuire (2007) notes that many older adults, especially the non-ambulatory, needed assistance evacuating before Hurricane Katrina ravaged New Orleans. Instead, many of them were left to fend for themselves. Some died while others had their primary disabling conditions untreated for days. Some went without prescribed medication, food and fluids while others were exposed to the elements. The need to evacuate disabled individuals is especially pertinent given that in the United States, 54% of those aged 65 and above have some form of disability (U.S. Census Bureau, 2001) and 20% have difficulty leaving their residences (Waldrop & Stern, 2003). About 32% of U.S. adults aged 70 and above indicate that they have difficulty with walking (McGuire, Ford and Ajani, 2006). The degree and severity of walking disability is high: 4% of adults aged 65 and above reporting the use of a wheelchair while 13% needed canes, crutches or walkers (U.S. Census Bureau, 2001). In a disaster, such individuals are among the most vulnerable groups (Saliba, Buchanan & Kington, 2004). They are thus likely to experience higher morbidity and mortality (Mokdad et al., 2005) due to difficulty when evacuating (Eldar, 1992; Fernandez et al., 2002).

For those who reside in long-term care establishments, the individual burden is less as facilities are legally responsible for evacuating them (Hardin, 2002). The facility decides whether to evacuate, arranges transportation, and plans appropriate temporary lodging. Long-term care facilities thus generally do not require as much assistance from emergency response personnel (Saliba et al., 2004). Moreover, long-term care institutions tend to support one another by lodging and caring for evacuated residents (Kuba et al., 2004; Saliba et al., 2004).

However, most older and disabled adults do not live in long-term care facilities; only 4% do (CDC & MIAH, 2004). This necessitates the home-based disabled and elderly and their families to plan for their evacuation (Eldar, 1992; Fernandez, Byard, Lin, Benson, & Barbera, 2002), including ensuring that the evacuation vehicle must accommodate the ambulatory equipment (Fernandez et al., 2002). Although FEMA recommends that people with disabilities form a self-help network of family, friends and neighbors to assist them during emergencies (FEMA, 2004), the extent of its success is unknown as older adults and people with disabilities often do not like to be identified for fear of becoming vulnerable to crime (IFAS, 1999) or are reluctant to leave their homes (Morrow, 1999). As such, the importance of evacuation efforts by the authorities, especially for the non-ambulatory, becomes increasingly apparent.

2. Humanitarian Logistics as a Supply Chain

Beyond practitioners, the field of humanitarian logistics has increasingly become a topic of interest to academics (Kovacs & Spens, 2007). Apte (2009) defines humanitarian logistics as a “special branch of logistics which manages [the] response supply chain of critical supplies and services with challenges such as demand surges, uncertain supplies, critical time windows in [the] face of infrastructure vulnerabilities and [the] vast scope and size of the operations.” Humanitarian logistics thus forms a large integral part of both disaster response and humanitarian relief (Kovacs & Spens, 2007; Thomas & Mizushima, 2005; Van Wassenhove, 2006), with logistics efforts accounting for 80% of disaster relief (Trunick, 2005). Although supply chains for humanitarian logistics are arguably among the “most dynamic and complex supply chains in the world” (Thomas, 2005), proper logistics preparation before a disaster strikes could better coordinate processes, technologies, and communications capabilities. This would improve the effectiveness and efficiency of the supply chains, and thus that of authorities’ response. Academic research, based on inputs from practitioners and using operations management and research analysis, could bridge the critical gap between logistical expertise and humanitarian relief.

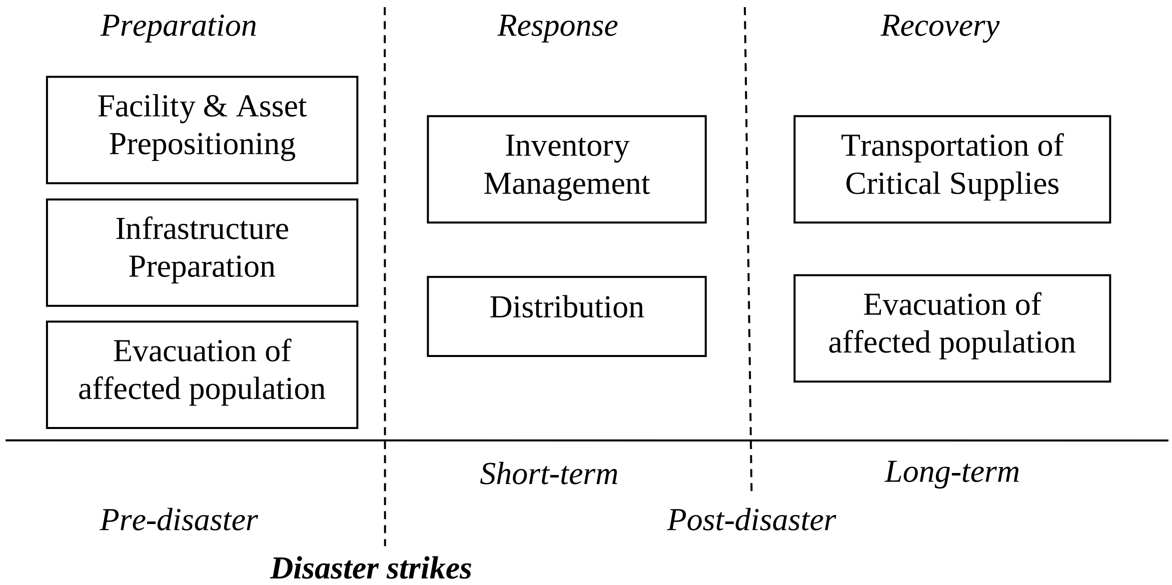


Figure 2. Timeline of humanitarian supply chain (After Apte (2009))

Adapting from Apte's (2009) model, the process flow in humanitarian logistics can be partitioned into three stages (Figure 2), although activities often straddle more than one stage: preparation before the disaster occur, immediate response after the disaster hit, and recovery in the post-disaster stage. In the first period, strategic prepositioning of facilities, assets and infrastructure preparations take place well ahead of a disaster. Prepositioning includes locating facilities, expanding storage, medical facilities, and shelters, while infrastructure preparation may include priming of airstrips. Pre-disaster evacuation of people from a target affected area helps to mitigate potential loss of lives. When the disaster happens, the immediate response includes collecting and distributing critical supplies to emergency coordinators in predetermined locations, with active management of the inventory. Finally, critical supplies and emergency personnel are transported in the last mile to specific affected zones, while evacuation of a wide range of injured individuals to temporary or permanent shelters and medical facilities is executed.

In particular, pre/post-disaster evacuation forms a significant component of relief operations. Evacuation can be defined as the "removal of residents as quickly as possible and with utmost reliability from a given area that has been considered a danger zone to safe locations" (Osman, 2010). Evacuation plans should be developed and rehearsed well in advance of disasters (Nisha de Silva, 2001). Nevertheless, evacuation planning is a complex problem which has several facets. These include the effects of different behavioral reactions and administrative factors (Dow & Cutter, 1998; Drabek, 1999; Perry, 1985; Vogt and Sorensen, 1992), the defining of evacuation zones (Sorensen, Carnes, & Rogers, 1992) and allocation of shelters (Sherali, Carter, & Hobeika, 1991).

Specifically on the determination of evacuation paths and schedules, there are two principle categories of evacuation situations: microscopic evacuation of buildings, ships and airplanes, etc., and macroscopic evacuation of whole cities or geographical regions (Hamacher and Tjandra, 2002; Lammel, Rieser, & Nagel, 2008). The former involves the evacuation of pedestrians, while the latter is associated with evacuation by vehicle. In the area of pedestrian evacuation, there has been considerable research in the last 20 years, e.g., Bakuli & Smith (1996) investigated design of building evacuation paths using extended queuing network models to improve throughput and total egress time. Excellent overviews of pedestrian evacuation models are provided by Schreckenberg & Sharma (2001), Galea (2003) and Gattermann, Waldau, and Schreckenberg (2006).

This thesis focuses on the second category, i.e., metropolitan-level evacuation. Under this context, evacuation can be further divided into two sub-categories: pre- and post-disaster evacuation (Figure 2.2) (Osman & Ram, 2011). The former focuses on precautionary evacuation, where comparison of evacuation time to hazard propagation time and associated risk can be conducted a priori. Hence, time and potential risks are the key components of this type of evacuation. The latter subcategory focuses on life-saving operations, i.e., route clearance and rescue of the injured. In both cases, efficient and effective evacuation modeling is needed to identify routes and schedules. Nonetheless, this thesis more specifically focuses on addressing the former sub-category of pre-disaster evacuation.

3. Evacuation Modeling

To resolve some of the traditional but highly complex issues described above, humanitarian logisticians and academics are increasingly relying on mathematical modeling to find the optimal solution, and increase the robustness of decision-making. For instance, initial work in humanitarian aid logistics attempted to locate emergency service facilities such as fire stations and ambulances using optimization models based on Set Covering Problems (SCP) and Facility Location Problems (FLP) (Cabot, Francis & Strary, 1970; Church & ReVelle, 1974; Fitzsimmons, 197; Shmoys, Tardos, & Aardal, 1997; Toregas, Swain, ReVelle, & Bergman 1971). For the critical supplies distribution and transportation problem, Rathi, Church & Solanski (1992) used linear programming formulations, while Sheu (2007) presented a hybrid fuzzy clustering optimization approach.

In the arena of macroscopic evacuation planning, literature research has focused on traffic assignment and evacuation departure scheduling (Ben-Tal, Chung, Mandalab, Yao, 2011), flow optimization, and classic ambulance routing (Parragh, 2009). Formulations of evacuation planning problems range from network flow models (Chiu, 2007; Cova and Johnson, 2003; Hoppe & Tardos, 2007), cell-transmission-models (Chiu, Villabos, & Gautam, 2007), traffic assignment models (Chiu & Zheng, 2007), multi-objective path selection models (Yuan & Wang, 2009), and transshipment models (Hoppe and Tardos, 2000). Optimization-based solution algorithms include those based on Capacity Constrained Route Planning (Lu, Huang, & Shekhar, 2003; Lu, George & Shekhar, 2005; Lu, 2006), Flip High Flip Edge (Kim & Shekhar, 2005), contraflow network reconfiguration (Shekhar & Kim, 2006), and Multi-Ant Colony Systems (MACS) (Zong, Xiong, Fang, & Li, 2010).

More realistic but complicating scenarios in the form of multiple commodities, customer priorities, and time-dynamic networks are occasionally considered. Haghani and Oh (1996) presented a large-scale multi-commodity, multi-modal network flow problem with time windows to transport a range of critical supplies using a vehicle fleet from depots to affected areas, while Barbarosoglu, Ozdamar & Cevik (2002) developed a mathematical model to efficiently plan crew/fleet configuration and flight routes for disaster relief helicopter missions. It aimed to achieve the multiple operational and tactical missions of determining: (a) the number of tours undertaken by each helicopter, (b) routing of helicopters from operation base to disaster area, (c) load/unload, delivery, transshipment and rescue plans of each helicopter in every tour and (d) refueling schedule of each helicopter at the operation base. Ozdamar, Ekinci & Kucukyazici (2004) further integrated time, solving a dynamic, time-dependent transportation problem during ongoing aid delivery. More recently, Yi and Ozdamar (2007) examined the problem of coordinating "transportation of commodities from major supply centers to distribution centers in affected areas and the transport of wounded people from affected areas to temporary and permanent emergency unit" and extended the earlier model as a mixed-integer, multi-commodity network flow problem treating vehicles as integer commodity flows in the first stage and providing schedules using a "vehicle splitting algorithm." The objective was to minimize delay in supplying critical commodities and health services. Chiu and Zheng (2007) presented a dynamic traffic assignment modeling technique based on a linear programming formulation of the cell transmission model to determine the optimal traffic assignment and departure schedule for multi-priority groups in response to a no-notice disaster. The objective was to minimize travel time over the entire system.

Nonetheless, several issues arise in attempting to model the thesis problem using such models and frameworks: they do not readily address the different nature of constraints or output forms, nor meet the objective of efficient algorithmic speed. For instance, Cova and Johnson (2003) did not provide the evacuation schedule, i.e., how many times a specific route can be used during evacuation and when to evacuate. Lu et al. (2005) presented a heuristic iterative algorithm Capacity Constrained Route Planner (CCRP) that finds the minimum time horizon that ensures 100% evacuation. However, resulting evacuation paths are not necessarily useful in practice because the evacuation paths from CCRP allow intersection nodes to hold flow for some periods of time, which is not possible in practice. In Hamacher and Tjandra (2002), the evacuation problem is formulated as a time-dynamic network flow optimization model, but its slow solution time is a major drawback of their approach for real-world large evacuation networks. In the dynamic transhipment problem (Herer & Tzur, 2001), a specific demand in each time period for each destination node is required, which is not always applicable for evacuation problems.

More critically, the abovementioned models do not take into account, nor can be readily adapted to address the thesis problem's complicating constraints in terms of heterogeneity of evacuee disability levels, multiplicity and heterogeneity of vehicle fleet and capacities, as well as the possibility of multiple tours. Evacuation planning often requires specificity and customization. A humanitarian logistics model that is capable of addressing the aforementioned multi-dimensional problem is, to the best of the authors' knowledge, non-existent. Given such inadequacies, the authors then turned to the established academic field of Vehicle Routing Problems (VRP), the subject of the next section, seeking to develop a more viable and tractable VRP-based formulation of the thesis problem.

B. VEHICLE ROUTING PROBLEMS (VRP) AND ITS VARIANTS: A METHODOLOGICAL REVIEW

Fundamentally, the VRP seeks the identification of an optimal set of routes to be performed by a fleet of vehicles, located in a depot(s), to fulfill the requirements of a given set of geographically-dispersed customers, subject to operational constraints. The objective is typically to minimize the global transportation cost (Bodin, Golden, Assad, Ball, 1983; Joubert, 2007) or distance travelled (Nagy & Salhi, 2005). The general VRP is often formulated as a graph-theoretic problem, with a set of vertices denoting originating depot, customer nodes and destination nodes, and an arc set with a non-negative cost associated with each arc between nodes.

The VRP and its variants form one of the most important classes of Combinatorial Optimization (CO) problems (Toth & Vigo, 2002). It has drawn tremendous interest from researchers because of its vital role in planning of distribution, transportation and logistics systems in sectors as diverse as bus routing, rubbish collection, mail and parcel delivery, food and beverage distribution, dial-a-ride taxi service, ambulance service, etc.

Besides road-based transportation applications, the VRP has also been seen in maritime and airlift planning.

The VRP was first introduced by Dantzig and Fulkerson (1954) and expanded by Dantzig and Ramser (1959). They described a real-world application concerning the delivery of petrol to service stations and proposed the first mathematical programming formulation and algorithmic approach. Since then, a wealth of variant models and solution approaches has been proposed for to obtain optimal and approximate solutions. Numerous commercial software that address various real-world VRPs are now available to industrial users.

1. VRP Variants

Beyond the basic VRP, numerous variants exist in the VRP family. These and their solution methods are discussed in several surveys by Solomon and Desrosiers (1988), Laporte (1992), Parragh, Doerner, and Hartl (2008), Eksioglu, Vural and Reisman (2009), Toth and Vigo (2002), as well as a 50th anniversary survey by Laporte (2009). A comprehensive taxonomy (Figure 3) shows how sophisticated and diverse the VRP literature is.

Given the vast number of possible scenarios, vehicles, customer requirement characteristics and constraints, it is practical to focus on the main and relevant variants for the purpose of this literature review. Broad classification schemes and naming convention have been given by Desrochers, Lenstra, and Savelsbergh (1990), Berbeglia, Cordeau, Gribkovskaia, Laporte's 3-field scheme (2007a; 2007b), Marinakis &Migdalas (2007), and Hosny (2010). In this thesis, we combine and adapt the schemes put forth by Toth and Vigo (2002) and Nagy and Salhi (2005) in Figure 4.

1. Type of Study

   1.1. Theory

   1.2. Applied methods

      1.2.1. Exact methods
      1.2.2. Heuristics
      1.2.3. Simulation
      1.2.4. Real time solution methods

   1.3. Implementation documented

   1.4. Survey, review or meta-research

2. Scenario Characteristics

   2.1. Number of stops on route

      2.1.1. Known (deterministic)
      2.1.2. Partially known, partially probabilistic

   2.2. Load splitting constraint

      2.2.1. Splitting allowed
      2.2.2. Splitting not allowed

   2.3. Customer service demand quantity

      2.3.1. Deterministic
      2.3.2. Stochastic
      2.3.3. Unknown [1]

   2.4. Request times of new customers

      2.4.1. Deterministic
      2.4.2. Stochastic
      2.4.3. Unknown

   2.5. On-site service/waiting times

      2.5.1. Deterministic
      2.5.2. Time dependent
      2.5.3. Vehicle type dependent
      2.5.4. Stochastic
      2.5.5. Unknown

   2.6. Time window structure

3. Problem Physical Characteristics

   3.1. Transportation network design

      3.1.1. Directed network
      3.1.2. Undirected network

   3.2. Location of addresses (customers)

      3.2.1. Customers on nodes
      3.2.2. Arc routing instances

   3.3. Geographical location of customers

      3.3.1. Urban (scattered with a pattern)
      3.3.2. Rural (randomly scattered)
      3.3.3. Mixed

   3.4. Number of points of origin

      3.4.1. Single origin
      3.4.2. Multiple origins

   3.5. Number of points of loading/unloading facilities (depot)

      3.5.1. Single depot
      3.5.2. Multiple depots

   3.6. Time window type

      3.6.1. Restriction on customers
      3.6.2. Restri ction on roads
      3.6.3. Restriction on depot/ hubs
      3.6.4. Restriction on drivers/vehicle

   3.7. Number of vehicles

      3.7.1. Exactly n vehicles
      3.7.2. Up to n vehicles
      3.7.3. Unlimited umber of vehicles

   3.8. Capacity consideration

      3.8.1. Capacitated vehicles
      3.8.2. Uncapacitated vehicles

   3.9. Vehicle homogeneity (in capacity)

      3.9.1. Similar vehicles
      3.9.2. Load-specific vehicles [2]
      3.9.3. Heterogeneous vehicles
      3.9.4. Customer-specific vehicles [3]

   3.10. Travel time

      3.10.1. Deterministic

4. Information Characteristics

   4.1. Evolution of Information

      4.1.1. Static
      4.1.2. Partially dynamic

   4.2. Quality of information

      4.2.1. Known (Deterministic)
      4.2.2. Stochastic
      4.2.3. Forecast
      4.2.4. Unknown (Real-time)

   4.3. Availability of information

      4.3.1. Local
      4.3.2. Global

   4.4. Processing of information

      4.4.1. Centralized
      4.4.2. Decentralized

5. Data Characteristics

   5.1. Data Used

      5.1.1. Real world data
      5.1.2. Synthetic data
      5.1.3. Both real and synthetic

   5.2. No data used

Notes

[1]Unknown refers to the situation in which information is revealed in real-time (dynamic and fuzzy studies fall under this category)

[2] Each vehicle can be used to handle specific types of load

[3] A customer must be visited by a certain type of vehicle

[4] Cost of operating a vehicle is not negligible

2.6.1. Soft time windows 2.6.2. Strict time windows 2.6.3. Mix of both 2.7. Time horizon 2.7.1. Single period 2.7.2. Multi period 2.8. Backhauls 2.8.1. Nodes request simultaneous pick ups and deliveries 2.8.2. Nodes request either linehaul or backhaul service, but not both 2.9. Node/Arc covering constraints 2.9.1. Precedence and coupling constraints 2.9.2. Subset covering constraints 2.9.3. Recourse allowed	3.10.2. Function dependent (a function of current time) 3.10.3. Stochastic 3.10.4. Unknown 3.11. Transportation cost 3.11.1. Travel time dependent 3.11.2. Distance dependent 3.11.3. Vehicle dependent [4] 3.11.4. Operation dependent 3.11.5. Function of lateness 3.11.6. Implied hazard/risk related	

Figure 3. Taxonomy of VRP literature (From Eksioglu, 2009; Honsy, 2010)

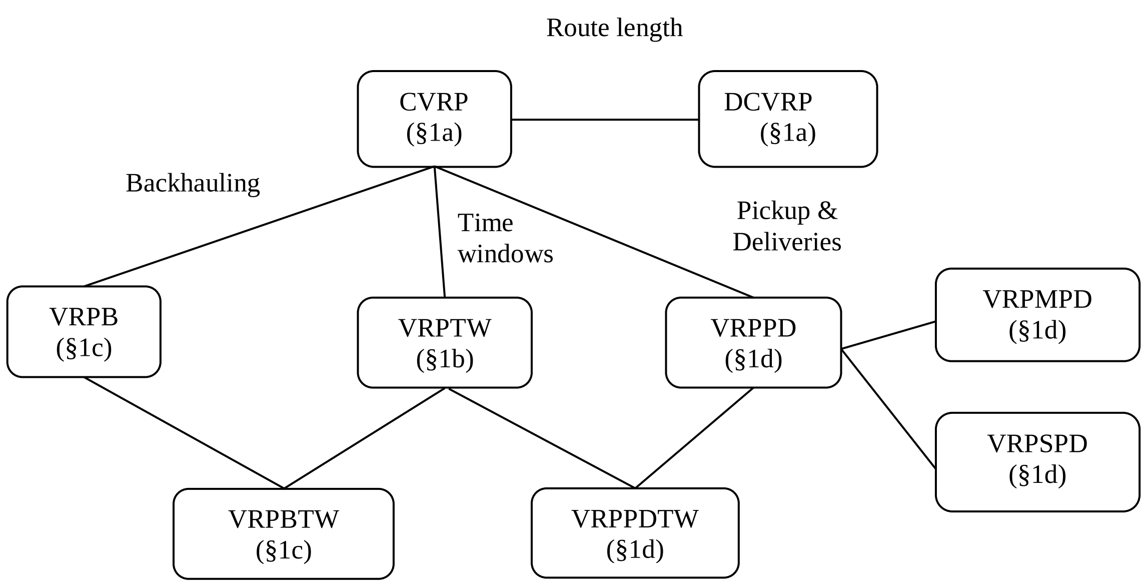


Figure 4. Broad classification scheme for VRP and its variants. Parentheses indicates which section headings under which further elaboration can be found. (After Toth and Vigo (2002), and Nagy and Salhi, 2005)

The following descriptions of the VRP variants in Figure 4 draw from Toth and Vigo (2002) and Nagy and Salhi (2005):

a. Capacitated VRP (CVRP)

The Capacitated VRP forms the simplest version of the VRP. Here, there exists a single depot, a set of identical vehicles with capacity constraints, and a set of customers who require delivery of goods from the depot. The objective is to minimize total cost (as a weighted function of number/length or travel time of routes), while subject to maximum traveling time and maximum capacity constraints on the vehicles. Each route must visit the origin depot, and serve each customer only once (Bodin, Golden, Assad, & Ball, 1983). Computationally, the CVRP is NP-hard (in the strong sense[1]), as it is a generalization of the related and well-known Traveling Salesman Problem (TSP) (Garey & Johnson, 1979; Mosheiov, 1994; Toth & Vigo, 2002). In the Distance Constrained VRP (DCVRP) variant, the capacity constraint is replaced by a maximum route length (or time) constraint. The objective is to minimize the total length or duration of the routes.

b. VRP with Time Window (VRPTW)

The VRP with Time Windows (VRPTW) is an extension of the CVRP in which capacity constraints are imposed and each customer is associated with a time interval called a time window. The time instants in which the vehicles leave the depot,

---

[1] A general computational problem may have numerical parameters. A problem is said to be NP-complete in the strong sense if it remains NP-complete even when all of its numerical parameters are bounded by a polynomial in the length of the input. A problem is said to be strongly NP-hard if a strongly NP-complete problem has a polynomial reduction to it. Nonetheless, in combinatorial optimization, the phrase "strongly NP-hard" is generally reserved for problems that are not known to have a polynomial reduction to another strongly NP-complete problem (Gary & Johnson, 1978).

the travel time for each arc and service time of for each customer are given. The service of each customer must start and end at pre-specified time instants. In case of early arrival at the customer node, the vehicle generally is allowed to wait until the service may start. The VRPTW is NP-hard in the strong sense, since it also generalizes the CVRP, arising when the time interval is infinite. The so-called TSP with Time Windows (TSPTW) is the special case of VRPTW in which there is only one vehicle.

c. VRP with Backhaul (VRPB)

The VRP with Backhauls (VRPB) is an extension of the CVRP in which the customer set is partitioned into two subsets. The first contains linehaul customers, each requiring a given quantity of product to be delivered. The second contains backhaul customers, where a given quantity of inbound product must be picked up. A precedence constraint exists: whenever a route serves both types of customer, all the linehaul customers must be served before any backhaul customer may be served. One reason for this is that it may be difficult to re-arrange delivery and pickup goods on the vehicles. Such an assumption makes implementation easier, since accepting pickups before finishing all deliveries results in a fluctuating load. This may cause the vehicle to be overloaded during its trip (even if the total delivery and the total pickup loads are not above the vehicle capacity), resulting in an infeasible vehicle tour. The VRPB is NP-hard in the strong sense, since they generalize the basic versions of the CVRP, arising when the backhaul subset is null. The case of VRPB in which time windows are present is called the VRP with Backhauls and Time Windows (VRPBTW) (Toth & Vigo, 2002).

d. VRP with Pickup and Delivery (VRPPD)

In the basic version of the VRP with Pickup and Delivery (VRPPD), each customer is associated with two quantities representing the demands of homogeneous commodities to be delivered and picked up at each customer. The main difference between VRPPD and the VRP is that customers may receive or send goods, while in the VRP all customers just receive goods from a depot (Nagy & Wassan, 2010). The VRPPD differs from the VRPB where the former involves transporting goods between any pickup and delivery locations while the latter sees goods being transported from a depot to linehaul customers and from backhaul customers to a depot. (Parragh, Doerner, & Hartl, 2008). In the basic version, it is assumed that, at each customer location, delivery is performed before pickup, i.e., there is a precedence constraint (Nagy & Salhi, 2005).

The objective is typically to minimize the total travelling distance or time cost while meeting customer demands. Other than the main constraint on vehicle capacity, others such as maximum distance or time windows may be present. Other variants of the VRPPD have also been introduced depending on, e.g., whether the origin/destination of the commodity is the depot or some customer location, whether one or multiple commodities are transferred, whether origins and destinations are paired, and whether people or goods are transported (Hosny, 2010).

The VRPPD is NP-hard, being a generalization of the classical VRP. The so-called TSP with Pickup and Delivery (TSPPD) is the special case of VRPSPD in which there is only one vehicle (Mosheiov, 1994). The case of VRPPD in which time windows are present has been studied in the literature and is called the VRP with Pickup and Deliveries and Time Windows (VRPPDTW) (Toth & Vigo, 2002).

Although the VRPPD is closely related to the OB-VRP, it has received considerably much less academic attention compared to the classical VRP and its main variants. While thousands of papers have been published for the latter, e.g., survey papers by Solomon and Desrosiers (1988), Laporte (1992), Eksioglu, Vural & Reisman (2009), Toth and Vigo (2002) and the survey paper in the 50th anniversary of the VRP by Laporte (2009), research on VRPPD is relatively scant (Savelsbergh & Sol, 1995). One contributing factor is the complexity of such problems and the difficulty in handling the underlying constraints.

Two key VRPPD models may be distinguished, briefly outlined below (Nagy & Salhi, 2005).

(1) VRP with Simultaneous Pickup and Delivery (VRPSPD)

The VRP with *Simultaneous* Pick-up and Delivery (VRPSPD) represents the case when no precedence constraints are imposed on the order in which the pickup and delivery must be performed (Bianchessi & Righini, 2007). Customers require not only the delivery of goods but also the simultaneous pick up of goods from them. A general assumption is that all delivered goods originate from the depot and all pickup goods must be transported back to the depot. Min (1989) first introduced this variant to solve a distribution problem of a public library, with the objective of minimizing the total travel distance/ time of the route by considering the vehicle capacity as the problem constraint.

(2) VRP with Mixed Pickup and Delivery (VRPMPD)

The VRP with *mixed* Pickup and Delivery represents the case where linehauls and backhauls can occur in any sequence on a vehicle route (Wade & Salhi, 2002). The VRPMDP can be considered the special case of the VRPSDP where either the delivery demand or the pick-up demand of each customer equals zero. Even though the VRPMDP is closely related to the VRPSDP, none of the solution approaches towards the VRPMDP can be applied directly for the strict VRPSDP, although some basic ideas can be transferred (Dethloff, 2001).

2. Relationships Between VRP Variants and Implications

The following relationships between VRP variants and their implications can be observed:

- Since the VRP is a generalization of the TSP, the relaxations and heuristics for the TSP are generally valid for the CVRP as well (Toth & Vigo, 2002);
- VRPSPD is a generalization of the VRPMPD (Nagy & Salhi, 2004). Thus, mixed and simultaneous VRPPD problems can generally be modelled using the same framework. Mixed problems can be thought of as simultaneous cases with either the pickup or the delivery load being zero; while the customers of simultaneous problems can be divided into pickup

and delivery entities to give a mixed formulation (Nagy & Salhi, 2005); and

- If all deliveries must be made before any pickups, the VRPMPD is reduced to the VRPB (Chen & Wu, 2006).

C. GENERAL SOLUTION APPROACHES

Three broad classes of solution approaches exist for solving CO problems such as VRPs: exact, heuristic and metaheuristic methods. In the literature, Laporte and Nobert (1987) presented an extensive survey that was entirely devoted to exact methods for the VRP, where they gave a complete and detailed analysis of the state-of-the-art up to the late 1980s. Other surveys also cover exact methods, but are often more devoted to heuristic and metaheuristic methods, including those by Christofides, Mingozzi, and Toth (1979), Magnanti (1981), Bodin et al. (1983), Christofides (1985), Laporte (1992), Fisher (1995), Toth and Vigo (1998), and Golden et al. (1998). Bibliographies were presented by Laporte and Osman (1995) and Laporte (1997), while books include those by Golden and Assad (1988) and Toth and Vigo (2002). Drawing from Hosney (2010) and Toth and Vigo (2002), this section will describe how the three broad classes and their main sub-classes work, as well as their intrinsic strengths and limitations.

1. Exact Methods

Until the late 1980s, the most effective exact approaches were inherited from the more extensive and successful work done for the exact solutions of the TSP, and have been further improved in recent years (Baldacci, Hadjiconstantinou, & Mingozzi, 2003). Exact methods will always identify an optimum solution for combinatorial optimization problems or VRPs, as long as a feasible solution exists and sufficient computational time is available. In general, the profitable exact algorithms trim down the solution space and number of different alternatives that need to be inspected in order to arrive at the optimum solution. The main exact algorithms that have been applied to solving CO/VRP problems are as described below.

a. Branch-and-Bound

First proposed by Laporte, Mercure, and Nobert (1987) to solve a basic Transportation Problem based on an Asymmetric VRP (AVRP), the Branch-and-Bound (B&B) algorithm is based on a systematic search of all possible solutions, discarding a large number of fruitless candidate solutions, i.e., “pruning,” using a depth-first strategy. The decision to reject a candidate is based on estimating upper and lower bounds of the quantity to be optimized, such that nodes whose objective function values are lower or higher than the current best are not investigated further. In the “branching” step, a given set of candidates is split into two smaller sets, while the “bounding” step computes upper and lower bounds for the function to be optimized within a given subset. The search terminates when all nodes of the search tree are either pruned or solved (Hosny, 2010).

The bounds are computed based on combinatorial relaxations of the constraints, e.g., spanning trees (Christofides, Mingozzi & Toth, 1981). Traditional basic relaxations are however generally unable to reach a solution quality sufficient for moderately-sized problems with 50 to 100 nodes (Toth & Vigo, 2002). More sophisticated and advanced bounds such as those based on Lagrangian relaxations have managed to increase the solvable problem size (Fisher, 1994; Toth & Vigo, 1997). The branch-and-bound method continues to be used widely in recent decades to solve the VRP and its chief variants. For many basic VRP variants, these algorithms still represent the state-of-the-art exact approaches available (Toth & Vigo, 2002).

b. Branch-and-Cut

Branch-and-Cut (B&C) is a B&B technique with an additional cutting step. This method has been very successful in finding optimal solutions of large instances of the closely related Symmetric TSP (STSP), as well as Prize Collected TSP (PCTSP) (Bérubé, Gendreau, & Potvin, 2009).

The concept is to decrease the search space of feasible candidates by adding new constraints (“cuts”). Adding the cutting step can improve the value returned in the “bounding” step, and allow solving of sub-problems without branching (Hosny, 2010). The cuts are based on linear relaxation of the constraint that variables have to be integers. If an optimal solution is not reached, the “branching” process decomposes the problem into two new problems, for example, by adding upper and lower bounds to a variable whose current value is fractional, as is done in branch-and-bound. Each new problem is then solved recursively using the same technique, where the optimal solution to the original problem will be the best of these two solutions. Such an amalgation of enumeration with cutting plane forms the core of the B&C method (Toth & Vigo, 2002).

Although the B&C method has been generally successful in solving many CO problems (Caprara & Fischetti, 1997), it may yield poor results if some of its components are weak, e.g., when (a) lack of a good algorithm to perform the cutting, (b) the number of iterations of the cutting plane phase is too high, (c) the linear program becomes unsolvable because of its size, or (d) the tree generated by the branching phase becomes too large and termination becomes unlikely within a reasonable time period. The most serious is problem (d) which can only be solved by strengthening the linear relaxation, i.e., adding linear inequalities that are satisfied by all solutions. Identifying such inequalities is non-trivial and requires a polyhedral study (Nemhauser & Wolsey, 1988) of the problem (Toth & Vigo, 2002). For a more extensive and in-depth explanation of the B&C technique, the reader is referred to Padberg and Rinaldi (1991), Thienel (1995), Jünger, Reinelt, and Thienel (1995), Caprara and Fischetti (1997), Jünger and Thienel (1998), and Toth and Vigo (2002).

c. Set-Covering-Based Algorithms

Set-Covering-Based (SCB) algorithms were first suggested by Balinski and Quandt (1964), for solving the CVRP based on a Set-Partitioning Problem (SPP) formulation of the VRP. Such a model has an exponential number of binary variables, each associated with a different feasible route. The SPP seeks to identify a set of routes with minimum cost, which serves each customer once and, possibly, satisfies additional restrictions.

The main advantage of this type method is that it allows for extremely general route costs, e.g., depending on the whole sequence of the arcs and on the vehicle type. Moreover, the additional side constraints need not take into account conditions on the feasibility of a single route. Hence, they can often be substituted by a compact set of inequalities, yielding a formulation whose linear programming relaxation is typically much tighter than that in other exact methods, i.e., linear relaxation of the SPP provides an optimal solution value very close to the optimal integer. Desrochers, Desrosiers, and Solomon (1992) reported an average relative gap between the optimal solution value to the linear relaxation and the optimal integer solution value of only 0.733% in the VRPTW case. Average-case gap analysis show that asymptotically, the gap tends to zero as the number of customers increase. Worst-case analyses for the related Bin Packing Problem (BPP), which can be viewed as a CVRP where all the customers are at the same location at a fixed (nonzero) distance from the depot performed found that lower bound is at least 75% of the value of the optimal integer solution (Chan, Simchi-Levi, & Bramel, 1995).

However, one of the main drawbacks is that even in loosely-constrained instances with tens of customers, billions of variables need to be managed. The explicit generation of all feasible routes (columns) is thus normally impractical, and one has to turn to a column generation approach to solve the linear programming relaxation of the model (Toth & Vigo, 2002). This was applied to a multiple-depot VRP by Ribeiro & Soumis (1994). Nonetheless, stabilized column generation is by itself an NP-hard challenge, and efficient approaches often rely on strong primal and dual components (du Merle, Villeneuve, Desrosiers, & Hansen, 1999). For a comprehensive description of the Set-Covering-Based method, the reader is referred to Agarwal, Mathur, and Salkin (1989) and more recent work by Hadjiconstantinou, Christofides, and Mingozzi (1995).

2. Classical Heuristics

Classical heuristics were mostly developed from the 1960s to 1990s, and can be broadly categorized into three classes: constructive heuristics that gradually build a feasible solution while tracking solution cost, two-phase heuristics that decompose the

VRP into its two natural components (clustering of nodes into feasible routes versus actual route construction), with possible feedback loops between the two stages, and improvement heuristics that upgrade feasible solutions by performing a sequence of arc or node exchanges within or between vehicle routes. The distinction between constructive and improvements methods however, is often blurred since most constructive algorithms incorporate improvements steps at some stage (Toth & Vigo, 2002). These classes will be briefly described below. In-depth overviews are provided by Nagy and Salhi (2005), Thangiah, Potvin and Sun (1996), Christofides (1985), Christofides, Mingozzi, and Toth (1979), and Bodin, Golden, Assad and Ball (1983).

a. Constructive Methods

The two main techniques to construct VRP solutions are savings and insertion heuristics. Savings heuristics merge existing routes using a savings criterion, whereas insertion heuristics gradually assign nodes to vehicle routes using an insertion cost.

(1) Savings-Based Algorithms

One of the most widely known VRP heuristics, Clarke and Wright's (1964) algorithm is based on the notion of savings. For the single-depot VRP, it begins with an initial allocation of one vehicle to each customer. It then uses a single vehicle to serve two customers on a single trip and computes the savings in total distance travelled. The larger the savings, the more desirable it becomes to combine the two nodes in a single tour. The savings are ranked and the node-pair is included into a route, so long as the resultant tour doe not violate any other constraints. Because of its simple manipulation of data, the Clarke-Wright algorithm runs very efficiently and can be applied to large problems. In addition, because nodes are added to routes one or two at a time, it is possible to check whether each addition violates any constraints, even when they are fairly complicated, e.g., a combination of maximum capacity, distance, time, and number of nodes that any vehicle can visit.

However, the solution generated by such a simple algorithm is not guaranteed to be close to the optimum. While experience has shown that the algorithm performs quite well most of the time, it is possible for pathological cases to yield very poor solutions (Toth & Vigo, 2002). To address this, Yellow (1970) proposed a more generalized savings which incorporates a route shape parameter. Other proposed enhancements compute matching-based savings based on TSP solution lengths of the two routes, using the savings values as matching weights, and merging the routes corresponding to optimal matchings, provided feasibility is maintained (Desrochers & Verhoog, 1989). In general, a matching-based algorithm yields better results than the classical Clarke and Wright method, but at the price of much longer computation time (Toth & Vigo, 2002). The memory requirements for sorting and ranking the savings list can also be very high for large problem sets, although this can be addressed via efficient heaping algorithms (Golden, Magnati & Nguyen, 1977; Nelson, Nygard, Griffin, & Shreve, 1985).

(2) Sequential-Insertion Heuristics

Fundamentally, a sequential insertion heuristic inserts an unrouted customer between two adjacent served nodes in a partially-finished route between depot and destination. In Solomon's (1987) seminal work, he categorized VRP tour-building algorithms into sequential versus parallel methods. The former constructs one route at a time until all customers are scheduled, whereas the latter simultaneously constructs routes, with the number of parallel routes either unconstrained, or constrained to a pre-specified number. When finding an initial solution, the initialization criterion finds the first customer (the seed customer) to insert into a route. A commonly-used initialization criterion is the farthest unrouted customer, and the customer with the earliest deadline. Once the seed customer has been identified and inserted, the algorithm considers, for the unrouted nodes, the insertion place that minimizes a weighted average of the additional distance and time needed to include a customer in the current partially constructed route, a step known as determining the insertion criteria. In the final step, the selection criteria tries to maximize the benefit obtained from inserting a customer in the current partial route rather than on a new direct route. If all remaining unrouted customers have no feasible insertion place, then a new route is initialized (Joubert & Claasen, 2006).

A shortcoming of Solomon's method is that it considers all unrouted nodes when calculating the insertion and selection criteria for each iteration, rendering the method computationally expensive. Nevertheless, with enhancements, they can become computationally efficient and operate in real-time, such that many of the commercial routing and scheduling software packages use insertion-based heuristics (Palmer, Dessouky, & Abdelmaguid, 2004). Furthermore, they are often the preferred method for generating an initial solution for improvement heuristics or meta-heuristics (Lu, 2005).

b. Two-phased Methods

Two-phase heuristics are divided into two classes: cluster-first, route-second methods and route-first, cluster-second methods. In the first case, nodes are first organized into feasible clusters, and a vehicle route is constructed for each of them. In the second case, a tour is first built on all nodes and is then segmented into feasible vehicle routes (Toth & Vigo, 2002).

(1) Cluster-First-Route-Second

Two-phase methods include the Sweep heuristic (Gillet & Miller, 1974; Wren & Holiday, 1971), the Generalized Assignment heuristic (Fisher & Jaikumar, 1981), and the Petal heuristic (Balinski & Quandt, 1964). The sweep algorithm applies to planar instances of the VRP. Feasible clusters are initially formed by rotating a ray centered at the depot. A vehicle route is then obtained for each cluster by solving a TSP. Some implementations include a post-optimization phase in which nodes are exchanged between adjacent clusters, and routes are re-optimized (Toth & Vigo, 2002). Fisher and Jaikumar's algorithm solves a Generalized Assignment Problem (GAP), rather than use a geometric method to form the clusters. It involves selecting seed nodes to initialize each cluster, allocating the customers to seeds by computing the cost of allocating each customer to each cluster, solving a GAP and solving a TSP for each cluster corresponding to the GAP solution. The Petal algorithm, first proposed by Balinski and Quandt (1964) as a natural extension of the Sweep algorithm, constructs a subset of feasible routes, called petals, and make a final selection by solving a set partitioning problem. However, it becomes impractical when the number of nodes is large. Agarwal, Mathur, and Salkin (1989) used column generation to solve small instances of the VRP optimally with number of nodes ranging from 10 to 25.

(2) Route-First, Cluster-Second Methods

First applied by Beasley (1983) to the VRP, route-first, cluster-second methods construct in a first phase a giant TSP tour for all customers, disregarding constraints. In the second phase, an arbitrary orientation of the TSP tour is chosen and the tour is partitioned into feasible vehicle routes according to capacity constraints. The process is repeated for several orientations and the best is chosen. Haimovich and Rinnooy Kan (1985) showed that if all customers have unit demand, this algorithm is asymptotically optimal, although this is seldom true in the real world. Little literature compares the computational efficiency of the route-first-cluster-second algorithm against other methods.

c. Improvement Heuristics

Improvement heuristics can either operate on each vehicle route taken on their own or on multiple routes at the same time (Toth & Vigo, 2002). In the single-route case, any improvement heuristic for the TSP can be applied. Most improvement procedures for the TSP can be described in terms of Lin's (1965) λ-opt mechanism, where λ arcs are deleted from the tour, and the remaining segments are reconnected in all possible permutations. If any profitable reconnection is found, it is implemented. The procedure stops at a local minimum when no further improvements are identified. In the multi-route case, procedures that exploit the multi-route structure of the VRP are developed to exchange arcs (Thompson & Psaraftis, 1993; Van Breedam 1994). Thompson and Psaraftis (1993) described a general b-cyclic, k-transfer scheme in which a circular permutation of b routes is considered and k customers from each route are shifted to the next route of the cyclic permutation. The authors showed that applying specific sequences of cyclic-transfer exchanges (with $b = 2$ or b variable, and $k = 1$ or 2) yields positive results. Van Breedam's improvement operations based on string cross, string exchange, string relocation, and string mix, can be viewed as special cases of 2-cyclic exchanges. He concluded that the string exchange yields the best overall improvement, albeit taking longer computational effort.

3. Metaheuristics

Metaheuristics are a recent class of approximate methods designed to solve hard CO problems arising in various different areas (Reeves, 1993). A metaheuristic iteratively guides a subordinate heuristic while performing a deep exploration of the most promising regions of the solution space. These methods typically combine sophisticated neighborhood search rules, memory structures, learning strategies and recombinations of solutions in order to efficiently find near-optimal solutions. Metaheuristics thus not only have the ability to continue the search beyond a local optimum where a heuristic would normally become trapped, but also be flexibly adapted to solve other optimization problems (Osman & Kelly, 1996; Toth & Vigo, 2002). Furthermore, although integer programming is commonly used to solve exactly most combinatorial problems, metaheuristics exploit the combinatorial nature of a problem rather than its integer programming formulation (Gendreau & Potvin 2005).

For overviews and more detailed treatment on metaheuristics, the reader is referred to Osman (1995), Osman & Kelly (1996), Aarts and Lenstra (1996), and Laporte & Osman (1996), Gendreau & Potvin (2005), Dreo, Petrowski, Siarry, & Taillard (2003), Glover & Kochenberger (2003). From a classification perspective, Gendreau & Potvin (2005) divided metaheuristics into two categories: single-solution metaheuristics, where a single solution (and search trajectory) is considered one at a time, and population metaheuristics, where multiple solutions evolve concurrently. Drawing from Gendreau & Potvin (2005), the next two sections will describe the metaheuristics under the two categories in detail. Nonetheless, an alternative classification can also be considered: primarily constructive metaheuristics, where a solution is built from scratch (through the introduction of new elements at each iteration), and improvement metaheuristics, which iteratively alter a solution. Constructive metaheuristics are mainly illustrated by the Greedy Randomized Adaptive Search Procedure (GRASP) and Ant Colony Optimization (ACO) while the rest are primarily improvement metaheuristics. That said, no classification scheme is perfect and metaheuristics do not always fall neatly into prescribed categories. For example, the GRASP methodology can contain an improvement phase to achieve local optimality through neighborhood search.

a. Single-solution Metaheuristics

Single-solution metaheuristics, generally considers one single solution (and search trajectory) at a time.

(1) Greedy Randomized Adaptive Search Procedure (GRASP)

Multi-start local search methods repeatedly apply a local search from different initial solutions. Using a fast greedy heuristic to generate starting solutions thus becomes desirable if the greedy solutions are sufficiently different to have a good sampling of local optima. In the 1980s, semi-greedy or randomized greedy heuristics (Feo & Resende, 1989) were proposed that added variability to greedy heuristics, leading to the search scheme known as GRASP (Gendreau & Potvin, 2005). A GRASP combines a greedy heuristic with randomization. Whenever the heuristic selects the next delivery to be inserted, it randomly picks from a pre-specified number of best choices. At each step of the construction heuristic, the elements not yet incorporated into the partial solution are evaluated with a greedy function, and the best elements are kept in a Restricted Candidate List (RCL). One element is then randomly chosen from this list and incorporated into the solution. This allows the algorithm to make choices that do not seem to be the best at the time but may provide better opportunities later. The heuristic is then executed multiple times and the best overall option is returned at the end (Toth & Vigo, 2002). Surveys on GRASP are provided in Festa and Resende (2002), and Resende and Ribeiro (2003).

The main shortcoming is that each restart is independent of previous ones, preventing exploitation of previously-obtained solutions to guide the search. Recent developments such as reactive GRASP (Prais & Ribeiro, 2000) address this by dynamically adjusting the size of the RCL based on the quality of recently-generated solutions.

(2) Simulated Annealing (SA)

Simulated Annealing (SA), first proposed by Kirkpatrick, Gelatt, and Vecchi (1983), has been applied to several types of discrete CO problems (Golden & Skiscim, 1986). It is a randomized local search procedure where a modification to the current solution leading to an increase in solution cost can be accepted with some probability. This algorithm is inspired by annealing of solids, where the energy of the system is minimized using slow cooling until the atoms reach a stable state. Slow cooling allows metal atoms to align and form a regular crystalline structure that has high density and low energy. In a CO context, the ground state of a material corresponds to the minimum energy configuration of its atoms, while the minimum energy configuration of a material corresponds to the minimum value of the objective function. At each iteration, the current solution is modified by randomly selecting a move from a particular class to some predefined cooling schedule, and a certain number of iterations are performed at each temperature level. SA accepts with certain probability feasible solutions which also increase the value of the objective function. Ideally, this acceptance probability should be close to one at a high temperature at the beginning of the cooling, and decreases to near-zero at a temperature close to zero near the end of the cooling. This prevents the SA algorithm from being trapped in a local minimum. In fact, unlike most metaheuristics, the SA method asymptotically converges to a global optimum (Aarts & Ten Eikelder, 2002; Henderson, Jacobson, & Johnson, 2003).

A critical issue of SA is determining an ideal annealing or cooling schedule. If the cooling rate is too fast, it would preclude the occurrence of the optimal solution. Recent advances from the basic SA have focused on using of different forms of static and dynamic cooling schedules with the intent to boost convergence speed without losing solution quality, and deterministic variants with threshold acceptance where a transition is accepted if it does not increase the cost by more than some predefined value; this value is progressively reduced as the algorithm unfolds (Dueck & Scheuer, 1990), and thermostatistical persistency, where different parts of the solution are locked as the search continues, due to frequent occurrences in previously-generated solutions (Chardaire, Lutton, & Sutter, 1995).

Overall, SA appeals to optimization problems in which obtaining a good solution, within a reasonable computational time, is preferred to an optimal solution with considerably longer solution time. Implementation is easy as it just requires a method for generating a move in the neighbourhood of the current solution, and an appropriate annealing schedule. It can also be used to tackle a wide range of CO problems, so long as an appropriate neighbourhood structure has been devised. Finally, high-quality solutions can be obtained, if a suitable neighbourhood structure and annealing schedule are picked (Hosny, 2010).

(3) Tabu Search (TS)

The principles of the TS method originates from the work of Glover (1986). TS, like SA, allows for intelligent exploration of the search space in an attempt to escape the trap of local optima. Nevertheless, there are three main differences between TS and SA (Hosny, 2010). Firstly, unlike SA, TS only accepts moves within the vicinity of the current solution that improve the objective function. Secondly, TS always searches for the best solution in the current neighborhood before applying the replacement criterion. Thirdly, the most distinguishing feature of TS is the use of a short term memory called a tabu list, in which recently visited solutions (or attributes of recently visited solutions) are stored to avoid short-term cycling where the search may be trapped within the boundaries of a certain neighbourhood region, oscillating among solutions that have been previously visited. Moves in the tabu list are prohibited and cannot be visited again for a certain number of iterations. In doing so, the algorithm is forced to explore new areas of the search space in order to escape local optima.

Typically, the search stops after a fixed number of iterations or a maximum number of consecutive iterations without any improvement to the incumbent (best known) solution.

Detailed overviews of TS can be found in Gendreau (2002; 2003), Glover (1989, 1990). Recent refinements to the basic algorithm introduce intensification mechanisms to enhance the search around good solutions, and diversification mechanisms to force the algorithm to explore new search areas. These are typically implemented via different forms of long-term memories (Glover & Laguna, 1997). Other developments include adaptive memories to both diversify and intensify the search, by taking different fragments of previously generated elite solutions and combining them to generate a new starting solution, similar to many population metaheuristics (Rochat & Taillard, 1995). Adaptive memories provide a generic framework for guiding local search and can be integrated with different types of metaheuristics. In Strategic Oscillation (SO) (Glover & Laguna, 1997), an oscillation boundary (usually, a feasibility boundary) is defined. The search is then allowed to go for a specified depth beyond the boundary before turning around. When the boundary is crossed again from the opposite direction, the search goes beyond it for a specified depth before turning around again. Repeating this procedure yields an oscillatory search pattern, whose amplitude can be tuned, e.g., tight oscillations favor a more thorough search around the boundary.

The reactive TS (Battiti & Tecchiolli, 1994) dynamically adjusts the search parameters based on the search history, where the size of tabu list is automatically changed when certain configurations occur too frequently to avoid short-term cycles. Wassan, Nagy, and Ahmadi (2008) controls the tabu tenure dynamically, and find initial solutions using a modification of the sweep algorithm. Nonetheless, the more recent implementations of TS are often cumbersome as a result of including multiple additional components that require many well-chosen parameters. Unified TS (Cordeau, Laporte, & Mercier, 2001) was a positive attempt to produce simpler, more flexible TS code with dynamic adjustment of (a few) parameters.

a. Population Metaheuristics

Population metaheuristics explicitly works with a population of solutions (rather than a single solution), by combining different solutions, implicitly or explicitly, in order to generate new solutions.

(1) Genetic Algorithms (GAs)

The idea of simulation of biological evolution and the natural selection of organisms dates back to the 1950s by early pioneers such as Alex Fraser (Fraser, 1957a; Fraser, 1957b). Nevertheless, the theoretical foundation of Genetic Algorithms (GAs) was established by Holland (1975). GAs are fundamentally inspired by how species evolve and adapt to their environment based on Darwin's theory of natural selection. In traditional GAs, each individual is usually represented by a string of bits analogous to chromosomes and genes, i.e., the parameters of the problem are the genes that are joined together in a solution chromosome. A fitness value is assigned to each individual in order to judge its ability to survive and breed (Hosny, 2010). A population of solutions from one generation generates the next generation of solutions through the application of operators that mimic those found in nature, i.e., selection of the fittest, crossover and mutation (Gendreau, 2005). By selecting the "fittest" parents, favorable characteristics spread throughout the population over several generations, and the most promising portions of the search space are tested. The population converges to an optimal or near optimal solution, such that the population evolves toward increasing uniformity, while its average fitness asymptotically approaches the highest possible fitness.

During the reproduction phase, a mating operator called crossover, combines the most desirable features from two selected parent solutions to create one or two offspring solutions. Not all selected pairs undergo crossover. A random choice is applied, with the likelihood of crossover assigned a given probability. If crossover is not performed, offspring merely duplicate their parents. This is repeated until a new population of offspring solutions is generated. Before replacing the old population, each member of the new population is subjected (with a small probability) to minute random perturbations via the mutation operator. Therefore, crossover allows a rapid exploration of the search space by producing large jumps, while mutation allows a small amount of random search. Starting from a randomly or heuristically generated initial population, this renewal cycle is repeated for a pre-specified number of iterations, and the best solution found is returned at the end. Detailed overviews of GA are provided in Beasley (2002), Michalewicz (1996), and Potvin (1996).

Fundamentally, a genetic algorithm is a randomized global search technique (Toth & Vigo, 2002). A pure GA uses little heuristic information about the problem domain, and hence can be applied to a wide range of ill-defined problems that do not lend themselves to specialized methods. It is noted that GA's success has often been achieved by departing from the traditional algorithm, e.g., the encoding of solutions into chromosomes is often either completely avoided (by working directly on the solutions) or specifically designed for specialized crossover and mutation operators. The distinctive feature of GAs remains the exploitation of a population of solutions and the creation of new solutions through the recombination of good attributes of two parent solutions. Many single-solution metaheuristics now integrate this feature, e.g., via adaptive memories (Reeves & Yamada, 1998)). Modern hybrid GAs further incorporate powerful local search operators as a form of mutation in order to address the situation where the population improves on average, but fails to generate near-optimal solutions. Rather than introduce small random perturbations into the offspring solution, a local search is applied to improve the solution until a local optimum is reached (Land, 1998; Moscato, 2002).

Overall, GAs are relatively adaptable, and can be easily hybridized to generate knowledge-augmented GAs. GAs can quickly reach fit individuals who are usually good enough as solutions to problems of a large magnitude. The main difficulty lies in designing an appropriate crossover operator, as combining two solutions rather than one is significantly more complex than developing a mutation operator or a simple neighbourhood move. This usually makes GAs implementation more difficult compared to more simple metaheuristics such as SA.

(2) Ant Colony Optimization (ACO)

ACO is based on how real ant colonies behave in order to find the shortest path between their nest and food sources (Çatay, 2006). Ants deposit pheromone on the routes they walk while seeking food. If other ants sense the pheromone, they are likely to follow that route rather than travel at random, thus reinforcing the route. As an increasing number of ants follow a particular route, the amount of pheromone on that route will increase, raising its selection probability by other ants. However, the pheromone evaporates over time, decreasing the probability of other ants following the route. The longer the route between the nest and the food source, the more the pheromone evaporates. Thus, the pheromone levels remain higher on the shorter paths. As a consequence, the level of pheromone laid is essentially based on the path length and the quality of the food source. In time, all ants are expected to follow the shortest path.

ACO simulates the natural behavior of real ants to solve CO problems by using artificial ants (Çatay, 2009). To apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants incrementally build solutions by moving on the graph using a stochastic construction process guided by artificial pheromone and a type of greedy heuristic information known as “visibility” (Dorigo, 2008). The amount of pheromone deposited on arcs is proportional to the quality of the solution generated and increases at run-time during the computation. Each time an element is selected by an ant, its pheromone level is updated by first removing a fraction of it, to mimic pheromone evaporation, and then by adding some new pheromone. When all ants have constructed a complete solution, the procedure is restarted with the updated pheromone levels. This is repeated for a fixed number of cycles or until search stagnation occurs.

While the basic Ant System (AS) solves small to moderate TSPs with comparable accuracy as other general-purpose heuristic approaches, e.g., genetic algorithms and simulated annealing, the simple ant algorithm is outrivaled by state-of-the-art specialized TSP algorithms for larger problems (Dorigo, Maniezzo, & Colorni, 1996). The insufficient optimization is due to (1) the best solution found can be lost by virtue of the probabilistic nature of route selection, (2) convergence is not guaranteed due to the uniform contributions of both the best and worst solutions to the pheromone updates, and (3) the collective memory of the colony stores unpromising variants, resulting in a considerable extension of the search area in larger problems (Shtovba, 2005)

Nevertheless, the original AS framework described in Dorigo (1992) provides inspiration for a number of enhancements that significantly improve performance. These extensions differ from the AS mainly in the way the pheromone update is performed and the pheromone trails are managed. Most are direct extensions of AS in the sense that they retain identical solution construction and pheromone evaporation procedures. They generally strongly exploit the best solutions found during the search and the most successful ones integrate explicit features to avoid premature stagnation of the search (Dorigo & Stutzle, 2004). The main differences between the various AS extensions lies in the techniques used to intensify or diversify the search process.

Some of the more prominent extensions include the Elite AS (EAS), rank-based AS, and Max-Min AS (MMAS). The idea of the EAS, first introduced in Dorigo (1992) and Dorigo, Maniezzo, and Colorni (1996), is to offer strong additional reinforcement to the arcs belonging to the best routes found since the start of the search. To circumvent slow convergence in the neighborhood of an optimum, elite ants deposit pheromones only on arcs of the best route found in order to attract more ants. The elitism ideas are further developed in rank-based AS (Bullnheimer, Hartl, & Strauss, 1999b) and MMAS (Stutzle & Hoos, 1997; 2000). In the rank-based AS, solutions found at each iteration are ranked such that bad routes are not retained. The pheromone amount deposited by an ant decreases with its rank; only the best (w-1) ants and one elite ant deposit pheromones. This can be visualized as w ants moving along the best route, (w – 1) ants moving along the best current route, (w – 2) ants moving along the second-best (by rank) route, etc. As such, the pheromone values on arcs of two routes of almost equal length can differ substantially, by at least 100/(w – 1)%. Therefore, in the neighborhood of the optimum, when the route lengths are almost the same, the ranking leads to a significant speed-up in searching for the best solution (Dorigo & Stutzle, 2004). As in the EAS, the best-so-far ant always deposits the largest amount of pheromone in each iteration to boost the probabilities of selecting the best route fragments.

In the MMAS, four main modifications are introduced vis-a-vis the AS. Firstly, it strongly exploits the best routes identified where only either the iteration-best ant, or the best-so-far ant is allowed to deposit pheromone. Unfortunately, such a strategy may run into stagnation where all the ants follow the same route, due to excessive growth of pheromone trails on arcs of a good, albeit suboptimal, route. To counteract this outcome, a second MMAS modification confines the pheromone trail values to an upper and lower bound. Thirdly, the pheromone trails are universally initialized to the upper pheromone trail bound, which, in conjunction with a small pheromone evaporation rate, widens the exploration of routes at the beginning of the search. Lastly, pheromone trails are reset each time the system approaches stagnation or when no improved tour has been generated for a certain number of consecutive iterations. (Dorigo & Stutzle, 2004)

The ACO algorithms described thus far achieved significantly better performance than the basic AS by introducing minor changes in the overall AS algorithmic structure. Some ACO algorithms that more extensively modify the features of AS and introduce new mechanisms based on ideas not found in the original AS have been proposed in literature. The Ant Colony System (ACS) (Dorigo & Gambardella, 1997) differs from the AS in three main areas. Firstly, it more strongly exploits the accumulated search experience through the use of a more aggressive action choice rule. Secondly, as in the EAS, pheromone evaporation and pheromone deposit occurs only on the arcs belonging to the best-so-far route, forcing the ants to search for an optimum in a narrow neighborhood of the previous best solution. Lastly, each time an ant uses an arc to move from node to node, it “eats” away some pheromone from that arc, reducing its attractiveness to other ants, to increase exploration of alternative routes. The solutions thus become more diverse owing to the dynamic update of the pheromone distribution (Dorigo & Stutzle, 2004; Shtovba, 2005).

The ACS was also the first ACO algorithm to use Restricted Candidate Lists (RCL), where a small list of preferential nodes that can be reached by an ant from a given node. The RCL contains a limited number of the best-rated nodes according to some heuristic criterion, e.g., increasing distances from the node an ant is on. Since such information is known a priori, candidate lists can be built before solving a problem instance and where they remain fixed during the entire solution process. An ant chooses a node outside the RCL only when the list has been exhausted. This allows exclusion of evidently unpromising variants and forces the ants to contemplate only the most promising routes, thus essentially reducing the search area. Experimental results have shown that use of candidate lists improves the solution quality and hastens the solution process, especially for larger problems (Gambardella & Dorigo, 1996; Dorigo & Stutzle, 2004).

Other ACO variations include the Hybrid Ant System (HAS) (Gambardella, Taillard, & Dorigo, 1999), where the pheromone trails guide a local search heuristic rather than a construction heuristic, the use of Multiple Ant Colony Systems (MACS) which interact by exchanging information about fruitful pheromone trails (Gambardella, Taillard, & Agazzi, 1999) and the exploitation of more sophisticated greedy heuristics to construct solutions (Le Louarn, Gendreau, & Potvin, 2004). Shtvoba (2005), Dorigo and Gambardella (1997), Dorigo, Maniezzo, and Colorni (1996) provide excellent overviews of the various ACO algorithms.

Overall, their versatility and robustness has led ACO algorithms to be popular in solving many types of CO problems, e.g., the TSP, the Quadratic Assignment Problem (QAP), and the job-shop scheduling problem (Dorigo, Maniezzo, & Colorni, 1996). It has also been applied to variants of VRPs (Doerner, K., Hartl, R. F., & Reimann, 2001). The idea of “attractiveness and “pheromone trails” are also exploited within other meta-heuristic techniques, e.g., in the crossover operator used in Zhao, Li, Sun and Mei (2008)’s genetic algorithm.

D. SOLUTION APPROACHES FOR RELATED VRP VARIANTS

This section focuses on the solution approaches researchers in literature have used to address the VRP variants that are more closely related to the OB-VRP, namely the VRPB, VRPSPD and VRPMPD. In contrast to the classic VRP and its other basic variants, literature is relatively scant for these variants (Toth & Vigo, 2002).

1. VRP with Backhaul (VRPB)

Exact optimization algorithms developed by Toth and Vigo (1997) and Mingozzi, Giorgi and Baldacci (1999) created two different mathematical formulations of the VRPB and managed to solve exactly problems with up to 100 customers. Toth and Vigo (1997) presented an integer linear-programming model and a branch-and-bound approach that uses a Lagrangian lower bound strengthened by adding inequalities in a cutting plane fashion. Mingozzi et al. (1999) presented an algorithm based on a new integer formulation, computing a valid lower bound by combining different heuristics for solving the dual linear-programming relaxation.

With regard to heuristics, Deif and Bodin (1984) modified the definition of the savings in the classical Clarke and Wright savings algorithm, while Goetschalckx and Jacobs-Blecha (1989) applied a space-filling-curve heuristic to obtain an initial solution, which was then improved by using a number of search heuristics. Goetschalckx and Jacobs- Blecha (1993) introduced a new heuristic based on the Generalised Assignment Problem for the formation of the clusters of customers that originate the routes. Halse (1992) adopted a cluster-first routing-second approach. In the first stage, the customers were assigned to vehicles, before a routing procedure based on 3-opt was performed, and a Lagrangean relaxation and column generation approach applied. A cluster-first route-second type heuristic is developed in which nodes are first distributed to vehicles and then the problem is solved using 3-opt algorithm. Solutions to problems with up to 100 customers for the VRPPD and 150 customers for the VRPB were reported. Toth and Vigo (1999) also presented a cluster-first-route-second heuristic where clusters are combined by solving an auxiliary assignment problem, using information provided by a proposed Lagrangian relaxation. The initial routes are built through a modified TSP heuristic. The final set of routes is then obtained through exchanging the intra-route, inter-route and outer-route arcs to improve the solution quality. More recently, Wade and Salhi (2002) developed an insertion algorithm for a particular situation in which pick-up and delivery can be partially mixed, i.e., collections may start as soon as a given fraction of deliveries has been completed.

With regard to metaheuristics, Duhamel, Potvin and Rousseau (1997) proposed a tabu search heuristic in which a greedy insertion procedure is used to obtain an initial solution. The initial solution is then improved through link or node exchanges. Osman and Wassan's (2002) tabu search method produces better average solutions than Toth and Vigo's (1999) algorithm, but requires much more computing time. They use two heuristics for generating the initial solutions: one that combines savings and insertion and another that combines savings and assignment. In their tabu method, the neighborhood is defined by the interchange of one or two consecutive customers between two routes. On the other hand, the tabu tenure is defined dynamically during the search by a reactive procedure. Brandão (2006) obtains a diversity of initial solutions from pseudo-lower bounds, an innovative feature for the VRPB. Further diversification was attained by the use of a random tabu tenure with consequent better results. Potvin, Duhamel and Guertin (1996) applied a GA to identify orderings that produce good routes and proposed a greedy route-construction procedure to insert customers one by one into the routes for a given ordering of customers. Wade and Salhi (2003) proposed an ant system algorithm for the mixed vehicle routing problem with backhauls.

2. VRP with Simultaneous Pickup and Delivery (VRPSPD)

Although a survey of the models and techniques for the VRPSPD can be found in Savelsbergh and Sol (1995), literature is generally lacking in contributions (Bianchessi & Righini, 2007). Most of the algorithms for solving the VRP-SDP are based on that of the classical VRP, with recent years focused on the development of heuristics and metaheuristics. Angelelli and Mansini (2002; 2004) solved the VRPSPDTW and VRPSPD using a branch-and-price strategy based on a set covering formulation. More recently, Lu and Dessouky (2004) propose a branch and cut based algorithm for the multiple vehicle version of the VRPSPD. Dell'Amico, Righini, & Salani (2006) presented an optimization algorithm based on column generation, dynamic programming, and branch and price method. However, the computational complexity of VRPSPD is evident from the computational result, in which one hour of computational time sometimes is not enough to solve a small size problem consisting of 40 customers.

In terms of heuristics, Casco, Golden and Wasil (1988) first examined basic constructive algorithms, improving on previous studies of Golden, Baker, Alfaro and Schaffer (1985), based on greedy insertion procedures and on the idea of savings introduced by Clark and Wright (1964). Min (1989) was the first to fully tackle the VRPSPD, solving a real-world problem faced by a public library, with one depot, two vehicles and 22 customers. The solution comprised three phases: clustering customer nodes, assigning vehicles to clusters, and creating the route of each vehicle by solving TSPs. The infeasible arcs were penalized (their lengths set to infinity), and the TSPs solved again. Mosheiov (1998) presented three greedy constructive algorithms based on tour partitioning to solve the problem with divisible demands, where each customer can be served by more than one vehicle. All three algorithms are route-first cluster-second: they accept as input a Hamiltonian tour (not including the depot) computed disregarding customer demands, and partition it into a set of subtours (Bianchessi & Righini (2007).

Salhi and Nagy (1999) proposed four insertion heuristics based on the methodology proposed by Golden et al. (1985) and Casco et al. (1988). They proposed a load-based insertion procedure that extends the idea of 1-insertion to cluster insertion. The basic steps of these heuristics are constructing partial routes for a set of customers, and then inserting the remaining customers into the existing route. These heuristic rules are mainly differentiated by the criteria for insertion and the number of customers per insertion. They tested four insertion methods (1-insertion, 2-insertion, connected-graph-cluster-insertion, and complete-graph-cluster-insertion) and concluded that the cluster insertion method offered positive improvement. The method can also be applied to the VRPB. Nagy and Salhi (2004) also proposed a local search heuristic with four phases and considered the degree of infeasibility. After finding an initial solution in the first phase, it is continuously improved in each of the following phases while maintaining a certain feasibility condition. In both papers, they addressed not only the VRPSPD, but also the VRPMPD where some customers require delivery while others require pickup. They showed that the VRPSPD is a generalization of the VRPMPD. In addition, they also extended the method for the multi-depot case. Dethloff (2001) and Dethloff (2002) presented insertion heuristics based on the concept of residual capacities and used four different criteria to solve the problem. in the insertion procedures proposed by Casco et al (1988), Salhi and Nagy (1999), and Dethloff (2001), the routes are all constructed sequentially. Shin (2009) proposed a novel 2-phase heuristic which consists of a clustering phase using the geometrical center of a cluster, and a route establishment phase applying a two-way search of each feasible route. Results showed that the suggested algorithm can generate better initial solutions for subsequent metaheuristics than other methods such as the giant-tour-based partitioning method or the insertion-based method.

In terms of metaheuristics, Bent and Henteryck (2006) proposed a simulated annealing approach for assigning customers to vehicles first with minimized number of routes and then using the Large Neighbourhood Search method to minimize the total travel cost. Tang and Galvao (2002) developed two local search heuristics based on Beasley (1983) and Gillet and Miller (1974). Tang and Galvao (2006) developed a TS metaheuristics which combines several techniques to obtain alternative inter-route and intra-route solutions, including relocation of a customer from one route to another route, interchanging a pair of customers between two routes, crossovering two routes, and 2-opt procedure. The VRPSPD was formulated to minimize the total traveled distance subject to maximum distance and maximum capacity constraints on the vehicles. Bianchessi and Righini (2007) proposed propose several kinds of heuristic algorithms for the VRPSPD with indivisible demands, mixed pick-ups and deliveries and both simple and composite demands. They compared four different tour-partitioning-based constructive algorithms, local search algorithms with various neighborhood structures, and TS algorithms. Their TS algorithm, based on complex and variable neighborhoods, combine arc-exchange-based and node-exchange-based neighborhoods, employing different and interacting tabu lists.

Genetic algorithms (Baker, & Ayechew, 2003; Christian, 2004) have drawn attention due to their robustness and flexibility. Cao & Lai (2007) proposed an improved genetic algorithm (IGA) using NOX crossover, swapping mutation and inversion operator were used as improved genetic operators to overcome the shortcomings of premature convergence and slow convergence of conventional genetic algorithm (GA). Cao (2008) and Cao, Lai and Nie (2008) proposed a hybrid algorithm based on combining the Differential Evolution (DE) theory (Storn and Price, 1996) and GAs. Çatay (2006) proposed an Ant Colony Optimization (ACO) algorithm introducing a new visibility function, producing comparable results to those of benchmark problems in literature. Gajpal and Abad (2009a; 2009b) proposed a MACS algorithm uses a new construction rule as well as two multi-route local search schemes for a VRPSPD.

3. VRP with Mixed Pickup and Delivery (VRPMPD)

There are also very few papers which explicitly deals with the VRPMPD. Being a close variation of the VRPSPD (Golden et al., 1985; Salhi & Nagy, 1999), most VRPMPD approaches are derived from the former. Similar to the VRPSPD, maintaining the feasibility of vehicle capacity is difficult in the VRPMPD since the available capacity fluctuates during the tour. Golden et al. (1985)'s approach is based on inserting backhaul (pickup) customers into the routes formed by linehaul (delivery) customers. Their insertion formula uses a penalty factor which takes into account the number of delivery customers left on the route after the insertion point. Casco et al. (1988) developed a superior load-based insertion procedure where the insertion cost for backhaul customers takes into consideration the load yet to be delivered on the delivery route (rather than the number of stops). Salhi and Nagy (1999) extended the insertion method of Casco et al. (1988) by allowing backhauls to be inserted in clusters, not just one by one. This approach yields some modest improvements and requires negligible additional computational effort. This procedure is also capable of solving simultaneous problems. Mosheiov (1994) investigated the TSPPD and demonstrates that if the solution is infeasible because some arcs are overloaded, feasibility can be attained by re-inserting the depot into the arc with the highest load. Anily and Mosheiov (1994) presents a solution method for the TSPPD by creating a minimum spanning tree. While its worst-case bound and computational complexity are better than that of Mosheiov (1994), its average performance is found to be slightly inferior. For the case of VRPSPB and VRPMPD with backhauls, Crispim & Brandão (2005) present a hybrid algorithm constructed through the use of TS and the variable neighborhood search metaheuristics.

4. Complicating Factors: Heterogeneous Vehicles and Multiple Depots

A number of further subvariants of the earlier VRP cases exist based on the multiplicity of the depots and the size and heterogeneity of the vehicle fleet. Nevertheless, literature is relatively scarce for such problems. A recent survey paper by Baldacci, Battara and Vigo (2008) reviews and compares variants of the CVRP involving heterogeneous fleet, their lower bounds and heuristic solutions.

Min, Current and Schilling (1992) was one of few pioneering research articles to address the multi-depot, multi-vehicle VRPB. They decomposed the model into three submodels/ phases: allocation of customers and vendors into clusters, assignment of customers and vendors to depots and routes, and individual route configuration. The decomposition procedure managed to solve a real-world problem with three depots, 134 customers and 27 vendors. Salhi and Rand (1993) improved the solution procedure by transforming the structure of the routes, by inserting or removing customers, and sometimes resulting in a reversal of the direction of parts of a vehicle route. This provides the foundation for the modification of the VRP algorithm into a VRPPD method and also for the operations to eliminate infeasibilities. For the multi-depot VRPSPD case, Nagy and Salhi (2005) proposed an integrated heuristic that established a weakly feasible solution first (one that checks only the total load delivered or picked up, but does not check vehicle capacity in-between nodes on the tour), and then remove infeasibilities through a combination of moves and an iterative procedure that reduces those infeasibilities in a controlled manner. Dondo and Cerda (2007) presented a novel three-phase heuristic/algorithmic derived from embedding a heuristic-based clustering algorithm within a VRPTW optimization framework based on MILP mathematical model, and efficiently solve case studies involving at most 25 nodes to optimality. To overcome this limitation, a pre-processing stage clustering nodes together is initially performed to yield a more compact cluster-based mixed integer linear programming problem formulation.

In the heterogeneous vehicle routing problem (HVRP), the number of available vehicles is fixed as a priori. The decision is how to best utilize the existing fleet to sever customer demands. The HVRP variant is first studied by Taillard (1999), and later by Tarantilis, Kiranoudis, & Vassiliadis (2003; 2004), Gencer, Top and Aydogan (2006) and Li, Golden and Wasil (2007). Gendreau, Laporte, Musraganyi and Taillard (1999) circumvent the tendency for local search technique to move towards a local optimum with the wrong fleet composition by diversifying the search by embedding within the algorithm a fleet change mechanism. To minimize total travel time, Ho, Ho, Ji and Lake (2008) developed two hybrid genetic algorithms (HGAs), where the first generates initial solutions randomly, while the second used the Clarke and Wright saving method and the nearest neighbor heuristic in the initialization procedure. They found that performance of the latter is superior in terms of the total delivery time.

In the Fleet Size and Mix VRP (FSMVRP), an NP-hard problem (Lenstra & Rinnooy Kan, 1981), the objective is now to find a fleet composition and corresponding route plan that minimizes the sum of routing and vehicle costs. Renaud and Boctor (2002) developed a sweep heuristic to find initial routes, and then solved a set partitioning problem to attain solutions to FSMVRP. Some tried to derive upper and lower bounds of the FSMVRP. Yaman (2006) gave several formulations of the FSMVRP with fixed cost, generalized subtour elimination and multistar inequalities. Based on derived valid inequalities, constraint lifting was used to improve the linear programming lower bounds. Choi and Tcha (2007) developed a set covering formulation and solved its linear relaxation by column generation to obtain the bounds. Comparing these methods on the Golden et al. (1984) benchmark instances, the best solutions have been obtained by Choi and Tcha (2007).

Matching-based saving algorithms were proposed by Desrochers and Verhoog (1991), Salhi and Rand (1993) and Osman and Salhi (1996). Salhi and Sari (1997) presented the idea of borderline customers for the multi-depot vehicle fleet mix problem, where two reduction tests enhanced the efficiency of a multi-level composite heuristic. The proposed heuristic is tested on benchmark problems involving up to 360 customers, two to nine depots and five different vehicle capacities. The heuristic yields solutions almost as good as those found by the best known heuristics but using only 5 to 10% of their computing time. Encouraging results were also obtained for the case where the vehicles have different capacities.

Meta-heuristics have also been applied by various authors. Bräysy, Dullaert, Hasle, Mester, and Gendreau (2008) applied deterministic annealing to the FSMVRPTW. Osman and Salhi (1996) developed a short-term memory TS using moves in 1-interchange neighborhood, while Gendreau et al. (1999) presented a TS algorithm embedded in an adaptive memory procedure. Taillard (1999) used heuristic column generation (HCG) where TS is first used to generate a set of good initial solutions; then an integer linear program, where each column in the program is a route from the initial solution, is solved to obtain the final solution. Wassan and Osman (2002) develop new variants of a TS meta-heuristic. These variants use a mix of different components, including reactive concepts, variable neighborhoods, hashing functions and special data memory structures, similar to adaptive memory procedures. Chen and Wu (2006) presented a hybrid heuristic based on an insertion-based procedure to generate good initial solutions and a heuristic based on the record-to-record travel, tabu lists, and route improvement procedures. Results showed that the proposed hybrid heuristic is able to reduce the gap between initial solutions and optimal solutions effectively for large-sized problems (50 to 199 nodes) and is capable of obtaining optimal solutions very efficiently for small-sized problem (15-20 nodes). Brandão (2009) developed a deterministic tabu heuristic, which restricts the moves in a nearest neighborhood whose size is determined by the estimated number of customers in a route. A GENIUS algorithm and giant tour are used to obtain initial solutions. Gendreau, Laporte and Semet (2001) proposed a dynamic model and parallel tabu search heuristic for real-time ambulance fleet relocation

Yi and Kumar (2007) applied the ACO metaheuristic to solve the logistics problem arising in disaster relief activities. The proposed method decomposes the original emergency logistics problem into two phases of decision-making, i.e., the vehicle route construction, and the multi-commodity dispatch. The sub-problems are iteratively solved. The first phase builds stochastic vehicle paths under the guidance of pheromone trails while a network-flow-based solver is developed in the second phase to assign different types of vehicle flows and commodities.

Ochi, Vianna, Drummond, & Victor (1998) develop a parallel GA hybrid with scatter search for the FSMVRP with fixed cost. A petal decomposition procedure is designed to build chromosomes. Each chromosome is a set of routes, multiple depots are used as route delimiters, and each route is optimized through GENIUS. A vehicle type is assigned to a customer by choosing the vehicle with the lowest product of its remaining capacity and its fixed cost. Prins (2004) applies an evolutionary algorithm to the CVRP and Wang, Golden and Wasil (2008) design a GA to solve the generalized orienteering problem. Liu, Huang, Ma (2009) designed different initial solution procedures, a new chromosome evaluation procedure and some local search moves that are specific to the FSMVRP, and proposed a new single parent crossover operator. Zhao, Mei & Sun (2009) introduced a pheromone-based crossover operator that utilizes both the local and global information to construct offspring. The local information used in crossover operator includes edge lengths and adjacency relations, while the global information is stored as pheromone trails. To improve the performance of genetic algorithms, a local search procedure is integrated into the GA, to act as a mutation operator. Lau, Chan, Tsui, & Pang (2010) proposed a Fuzzy Logic Guided Genetic Algorithm (FLGA) to solve the problem. The role of fuzzy logic is to dynamically adjust the crossover rate and mutation rate after ten consecutive generations, for the problem in which multiple depots, multiple customers, and multiple products are considered.

E. CONCLUSIONS FROM LITERATURE REVIEW

In summary, this chapter describes the research from both thematic and methodological perspectives over the last 60 years. A number of salient conclusions can be drawn from the literature review.

From a problem definition perspective, the thesis problem is fundamentally different from past research in both humanitarian logistics and VRP fields. In particular, the OB-VRP is different from the classic VRP and its basic variants in three ways:

- Problem complexity. The OB-VRP possess complicating constraints in terms of ranked heterogeneity of customer demand (evacuee disability levels), multiplicity and heterogeneity of vehicle fleet and capacities, as well as the possibility of multiple tours by any given vehicle, which is a key issue. Such an overburdened problem has not been dealt with in literature, and existing models are not readily adaptable for the OB-VRP.
- Objective function. The objective of the OB-VRP is fundamentally different from that of VRP models in literature, which almost exclusively use distance and/or vehicle cost as the objective function. While routing cost could have been retained in the objective function, the aim in disaster evacuation is to save lives. As such, the authors have set the objective as the minimizing of the number of un-served customers. The implications are that a simple swap of customer nodes, a typical heuristic/metaheuristic operation no longer improves the objective function: to improve the objective function, an un-served customer must be added to one of the routes, thereby increasing time. Reducing time, number of vehicles, or number of routes does not improve the solution as it does in traditional VRPs. Traditional solution approaches may thus not be suitable or directly adaptable for the OB-VRP.
- Problem size. Despite advances in algorithmic approaches, solving the VRP, in particular the intricate VRPPD subvariants, to optimality remains difficult for very large problem sizes. Exact algorithms that may be used to provide optimum problem solutions cannot solve VRP instances with more than 50-100 customers (Hasler & Kloster, 2007). Approximation methods are able to solve problems with hundreds of requests, e.g., Toth and Vigo (1997) showed their approach to be computationally viable for a

problem consisting of more than 300 requests. Nonetheless, larger realistic instances are often solved to optimality only in specific scenarios (Toth & Vigo, 2002).

From a solution perspective:

- Difficulty in comparing solution methods. The relative evaluation of competing approaches is difficult, because a benchmark problem set has not been developed for the VRPPD as it has for the generic VRP or VRPTW. The primary reason is the plethora of problem variants that the literature has addressed. Generally, much of the work has stemmed from applications that induced modeling differences. For example, the manner in which service quality is represented in the objective function or the constraints is often situation-specific. Furthermore, data sets currently used as benchmarks are made up of instances that are too small to allow one to differentiate sharply between the various implementations of some of the metaheuristics, in particular for TS (Toth & Vigo, 2002).

- Heuristics versus metaheuristics. Most standard construction and improvement procedures in use today are heuristics. They perform a relatively limited exploration of the search space and typically produce good quality solutions within modest computing times, and hence are still widely used in commercial packages. In metaheuristics, the emphasis is on performing a deep exploration of the most promising regions of the solution space. Solution quality is much higher than that from classical heuristics, but the expense is increased computing time. Moreover, the procedures usually are context-dependent and require finely-tuned parameters, which may make their extension to other situations difficult (Toth & Vigo, 2002).

- Solution complexity. Recent research tends to exhibit complexity and convolution in solution methodology. This is due to the need to handle the difficult, and sometimes conflicting, problem constraints, especially for the VRPPD subvariants. Approaches often adopt problem-specific techniques or hybridize several heuristics and metaheuristics, or utilize heuristics within exact methods in order to obtain good-quality solutions. Unfortunately, such complex composite techniques also means it becomes challenging to assess which algorithmic component has contributed most to the success of the overall approach, if indeed, necessary at all. Furthermore, due to the many problem constraints that often apply, obtaining a feasible solution itself may be a challenge. Since the generation of infeasible solutions cannot be easily avoided during the search, solution techniques often add a repair method to fix infeasibility. This inevitably renders the solution algorithm less elegant and slows down the optimization process.

THIS PAGE INTENTIONALLY LEFT BLANK

III. PROBLEM DEFINITION AND MODEL FORMULATION

This chapter describes the Overburdened Vehicle Routing Problem (OB-VRP) in Section A. Section B gives the OB-VRP a formal definition as a graph theoretic model, while Section C provides a linear mixed integer formulation first presented in Apte and Heath (2011).

A. PROBLEM DESCRIPTION

The OB-VRP takes place in the context of a sudden-onset disaster where roads are still traversable. The aim is to develop a routing plan that sends vehicles from depots to pick up and evacuate as many mobility-challenged evacuees as possible from their homes to a common shelter, within given constraints. Known a priori are the number, location, disability level, and loading/unloading times of evacuees, the number and location of all depots, as well as the fleet type and size. The mapping of evacuees' disability level to the minimum type of vehicle required is also known; lower disability severity evacuees can be transported on a vehicle designed for higher severity people, but not vice versa. Loading and unloading times vary for different evacuees. There is no prioritization of people during evacuation. While evacuees are characterized by their location, not all evacuee locations are unique (it is possible to have more than one evacuee at a location. It is also not assumed that a vehicle has to pick up everyone at the same location simultaneously. Multiple trips are allowed so long as the overall evacuation time available is not exceeded.

The vehicles may be based at more than one originating depot, though some vehicles may have the same originating depot. The capacities of each vehicle for each need level of customer is known. The total load carried by each vehicle at any given time cannot not exceed its capacity. Each vehicle type takes the same amount of travel time to travel from point to point. While there is no limiting starting and ending times in place, the entire evacuation must take place within one time window constrained by total available time.

B. PROBLEM DEFINITION

The OB-VRP may be defined as the following graph theoretic problem. Let G = *(V*, A) be a connected and complete graph, where $V = \{0,..., N+K\}$ is the node set and A is the arc set. Nodes $i = 1,..., N$ correspond to the customers (evacuees), whereas node 0 corresponds to the common shelter node s and nodes $N+1, \ldots, N+K$ are the origin depot nodes associated with the K vehicles.

A nonnegative cost, t_{ij}, is associated with each arc $(i, j) \in A$ and represents the time cost spent to go from node i to node j. Although the use of loop arcs (i, i) is not allowed, if some evacuee nodes have the same physical location, this will be denoted by an arc travel time of zero between them but they remain separate nodes. G is a non-directed graph and the cost matrix t is symmetric, i.e., $t_{ij} = t_{ji}$. The time-cost matrix satisfies the triangle inequality, i.e., it is not convenient to deviate from the direct Euclidean path between two nodes i and j.

Each customer node i is associated with a known non-negative demand d_{il} where l denotes the disability severity level of that customer node. u_i characterizes the unloading and loading time for customer i. Both d_{il} and u_i are deterministic and known in advance.

A set of K heterogeneous vehicles, each with capacity c_{kl}, is available at the K origin depot nodes. To ensure feasibility, it is assumed that $d_{il} \leq c_{kl}$, for each $i = 1, \ldots, N$. Each vehicle may perform more than one tour by dropping at the shelter node customers picked up earlier and returning for additional customers. R represents the maximum number of tours vehicles can make, while T specifies the total time available to perform the evacuation.

The OB-VRP then aims to determine the feasible assignment of vehicles to people, and the corresponding sequence of tour-routes, so that the objective function (total number of customers not evacuated) is minimized, such that:

- each tour-route visits the shelter node;
- each customer node is served at most once;

- the sum of the demands of the customer nodes visited by a vehicle on any one tour must be nonnegative and never exceeds its capacity c_{kl}; and
- total time taken does not exceed time available for evacuation T.

C. MODEL FORMULATION

Three different basic modeling approaches have been proposed for the VRP and its variants in the literature. The first approach uses vehicle flow formulations which use integer variables, associated with each arc of the graph, to count the number of times the arc is traversed by a vehicle. Vehicle flow formulations are well-suited for cases where relevant constraints can be effectively modeled through an appropriate definition of the arc set and of the arc costs. That said, vehicle flow models cannot be used to handle cases when the solution cost depends on the overall node sequence (Toth & Vigo, 2002). The second approach uses models based on commodity flow formulation, where additional integer variables are associated with the arcs and represent the flow of commodities along the paths traveled by the vehicles. Such models have been used in recent times as a basis for exact VRP solutions. The third approah formulates the VRP as a Set-Partitioning Problem (SPP). The main advantage of this model approach is that that it produces a formulation whose linear programming relaxation is typically much tighter than in the other methods (Toth & Vigo, 2002); however, such models generally require dealing with a very high number of variables compared to the others.

We now provide a four-index vehicle flow formulation, first presented in Apte and Heath (2011). Two-index vehicle flow models have already been used extensively to model basic VRPs but they generally are inadequate for more complex variants. In fact, they can be used only when the cost of the solution can be expressed as the sum of the costs associated with the traversed arcs. In addition, it is not possible to directly know which vehicle traverses an arc used in the solution. Hence, these models are not suited for the cases where the cost (or the feasibility) of a circuit depends on the overall vertex sequence or on the type of vehicle allocated to the route. The three-index vehicle flow formulation somewhat overcomes this drawback, explicitly indicating the vehicle that traverses an arc, so that more involved constraints may be imposed on the routes.

Nonetheless, the complexity of the OB-VRP requires the introduction of a fourth index to represent the trip or tour sequence, thus generalizing and strengthening the lower-dimensional formulations.

1. Sets

A set of all arcs in graph

V set of all nodes in graph, $V = \{0,...,N+K\}$

O subset of V consisting of all origin depot nodes for vehicles; within set V, O will be indexed with N+1,...,$N+K$ where 1,...,K is the vehicle originating at node N+1,...,$N+K$, respectively

s shelter node which is the destination for all customers; within set V, s will be indexed with 0

C subset of V consisting of all nodes in V with a customer needing to be evacuated; each node in C represents one person needing to be evacuated (1 unit of supply); within set V, C will be indexed with 1,...,N

C^+ union of s and C; the set of all possible nodes vehicles may visit once they leave their depot

2. Parameters

T total time available to perform the evacuation

K number of vehicles

N number of customers needing to be evacuated

R maximum number of trips a vehicle can make

L number of levels of different transportation needs customers can have

t_{ij} time it takes for any vehicle to traverse arc (i, j) $\forall\ i,j \in V$

d_{il} supply at node i, with need level l, $\forall\ i \in C,\ l \in L;\ \sum_{l \in L} d_{il} = 1, \forall i \in C$

c_{kl} capacity of vehicle k for customers of level l, $\forall\, k \in \{1,...,K\}$, $l \in L$

u_i unloading and loading time for customer i, $\forall\, i \in C$

3. Variables

x_{ijkr} equals 1 if arc (i, j) is traversed by vehicle k on trip r in the solution, $\forall\, i,j \in V$, $k \in \{1,...,K\}$, $r \in \{1,...,R\}$; equals 0 otherwise

y_{ikr} equals 1 if customer i is serviced by vehicle k on trip r, $\forall\, i \in C$, $k \in \{1,...,K\}$, $l \in L$, $r \in \{1,...,R\}$; equals 0 otherwise

4. Objective Function

Minimize: $$N - \sum_{k \in K} \sum_{r \in R} \sum_{i \in C} y_{ikr} \tag{3.1}$$

5. Constraints

Subject to:

$$\sum_{j \in C^+} x_{ijk1} = 1 \qquad \forall i = N + k, k = 1,..., K \tag{3.2}$$

$$\sum_{j \in O} x_{ijk1} = 0 \qquad \forall i = N + k, k = 1,..., K \tag{3.3}$$

$$\sum_{j \in V} x_{j0k1} = 1 \qquad \forall k = 1,..., K \tag{3.4}$$

$$\sum_{j \in V} x_{ijk1} = \sum_{j \in V} x_{jik1} \qquad \forall i \in C, k = 1,..., K \tag{3.5}$$

$$\sum_{k \in K} \sum_{j \in V} x_{ijkr} = 0 \qquad \forall i \in O, r = 2,..., R \tag{3.6}$$

$$\sum_{j \in V} x_{ijkr} = \sum_{j \in V} x_{jikr} \qquad \forall i \in V, k = 1,..., K, r = 2,..., R \tag{3.7}$$

$$\sum_{j \in V} x_{ijkr} = y_{ikr} \qquad \forall i \in C, k = 1,..., K, r = 1,..., R \tag{3.8}$$

$$\sum_{k=1}^{K} \sum_{r=1}^{R} y_{ikr} \leq 1 \qquad \forall i \in C \tag{3.9}$$

$$\sum_{i \in C} y_{ikr} d_{il} \leq c_{kl} \qquad \forall k = 1,...,K, r = 1,...,R, l = 1,...,L \qquad (3.10)$$

$$\sum_{i \in C} \sum_{j \in C} x_{ijkr} \leq |S| - 1 \qquad \forall S \subseteq C, |S| \geq 2, k = 1,...,K, r = 1,...,R \qquad (3.11)$$

$$\sum_{r=1}^{R} \left(\sum_{i \in C} 2u_i y_{ikr} + \sum_{i \in V} \sum_{j \in V} t_{ij} x_{ijkr} \right) \leq T \quad \forall k = 1,...,K \qquad (3.12)$$

$$x_{ijkr} \in \{0,1\} \qquad \forall i \in V, j \in V, k = 1,...,K, r = 1,...,R \qquad (3.13)$$

$$y_{ikr} \in \{0,1\} \qquad \forall i \in C, k = 1,...,K, r = 1,...,R \qquad (3.14)$$

The goal in Equation (3.1) is to minimize the total number of customers that do not get evacuated. Constraints (3.2) and (3.3) ensure that each vehicle leaves its own depot exactly once on the first trip and does not go to another vehicle’s depot. Constraint (3.4) ensures that each vehicle enters the shelter once on its first trip (having departed from its depot on the first trip). Constraint (3.5) is the balance of flow constraint for the customer nodes for the first trip of each vehicle. Constraint (3.6) ensures that no vehicles leave any of the depots on subsequent trips. Constraint (3.7) is the balance of flow constraint for all nodes on subsequent trips. Constraint (3.8) sets the value of the *y* variables.

Constraint (3.9) ensures each customer is serviced no more than once. Constraint (3.10) is the capacity constraint for each customer need level *l*, for each trip that each vehicle makes. Constraint (3.11) is the classic subtour elimination constraint, but note that sub-tours are only infeasible if they occur entirely within the subset of customer nodes. Constraint (3.12) constrains the total of all time spent by each vehicle loading customers, unloading customers, and traveling to be within the total time available for evacuating. Constraints (3.13) and (3.14) define the variables.

An important issue is the definition of objective. This is a difficult, and to some extent, unresolved task in humanitarian logistics given the intricate difficulty in assigning a suitable performance metric. Should the objective cost be based on money spent and/or cost to the economy? Or should it be based on the cost the fatalities or suffering of the survivors or both? Assigning monetary value to social utility is a complex and ethical issue. Nonetheless, over the years, the objective of evacuation research has somewhat evolved from minimizing costs to maximizing public welfare (ReVelle, Bigman, Schilling, Cohon, & Church, 1977). This is expected given that the primary aim in a supply chain in humanitarian logistics is to minimize “loss of life and alleviate suffering” (Thomas, 2003). The supply chain for any humanitarian response can be deemed truly successful only if it “mitigates the urgent needs of a population with a sustainable reduction of their vulnerability in the shortest amount of time and the least amount of resources” (Van Wassenhove, 2006, p. 480). As such the OB-VRP has adopted the objective of minimizing the number of un-served customers (evacuees).

THIS PAGE INTENTIONALLY LEFT BLANK

IV. SOLUTION DEVELOPMENT

A solution algorithm was developed with emphasis on several priorities that would best benefit an emergency responder. First, the algorithm needed to be robust enough to always be able to find some solution. Second, a simple algorithm that could find a relatively good solution was desired over a complex one that could find the absolute optimum solution, since the ultimate goal is a "push-button" algorithm that a non-academic user could operate quickly and effectively in a disaster situation.

A. PRELIMINARY STUDIES

The early search for a suitable algorithm centered on basic heuristic methods such as an adaptation of the Clarke-Wright Savings algorithm (Clarke & Wright, 1964). The primary advantage of these methods is their sheer simplicity; however, in adapting these methods to the OB-VRP, they became unnecessarily complex, without any additional improvement in accuracy.

A search through modern metaheuristics revealed several promising candidates, including Genetic Algorithms, Simulated Annealing, Tabu Search, GRASP, and Ant Colony Optimization (ACO).

The ACO appeared particularly suited to the problem: it could be implemented easily and it allowed a simple route-construction heuristic. In addition, it could be expanded with additional routines such as a Multiple Ant Colony System or the incorporation of local search heuristics within the route construction module. Finally, ACO routines have been shown in literature (Dorigo, Maniezzo, & Colorni, 1996) to perform on par with other popular metaheuristics such as Tabu Search, strengthening confidence that the ACO can find a very good solution.

B. SOLUTION FEATURES AND INSIGHTS

The final algorithm merged an ACO routine with two different greed-based heuristics. The ACO routine will be discussed in Section 1, and the heuristic contributions will be discussed in Section 2.

1. Ant System

Dorigo & Stutzle (2004) and Shtovba (2005) each reviewed the various incarnations of the ACO; based upon a review of their work, we built a hybrid Ant System (AS) algorithm designed to take advantage of the best features of each.

First, the base system was modeled after the Min-Max Ant System (MMAS) of Stutzle & Hoos (1997), predominantly because of that algorithm's particularly strong evaluated ability to quickly find good quality solutions. The MMAS differs from the original AS algorithm by specifying an upper and lower bound on the pheromone levels. The pheromone levels are universally set to the maximum for the initial iteration. Only the best solution at the end of each iteration receives pheromone deposition. Since the OB-VRP is a new problem, the algorithm was initially required to evaluate a large solution space, to avoid finding local optimums; to this end, the MMAS was modified to allow several good solutions to affect the pheromone levels, as opposed to the original model that only considered the best ant per iteration.

This was effected by creating a "best solutions" list, and adding elite ants (Dorigo, Maniezzo, & Colorni, 1996) and a ranked contribution system (Bullnheimer, Hartl, & Strauss, 1999b). The elite ant is a concept borrowed from one of the first evolutions of the original AS algorithm (Dorigo, 1992); the original AS allowed pheromone deposition for each ant's constructed route, and the elite ants were an additional weighting of pheromone deposition for the best route, analogous to σ extra ants depositing their pheromones on that route.

The MMAS as originally constructed restricted pheromone deposition to one elite ant; however, Çatay (2009) used a rank-based system with an MMAS. The ranked system is a simple concept similar to the elite ant. Instead of only the best solution receiving deposited pheromones, all the solutions in the "best solutions" list receive some pheromone deposition. The top-ranked solution in the list receives a higher proportion of pheromones than the lowest-ranked solutions in the "best solutions" list. That is, for a "best solutions" list of size *w*, the ranked system is equivalent to *w* ants marching over the best solution, *w*-1 ants marching over the second best solution, and so forth.

Each of these contributing factors was controlled by a set of variables which were included in the initial call of the routine; at any time, they could be turned on or off, or set to different levels. For example, the initial call could prescribe 10 elite ants, plus a ranked system including the top five solutions. In this particular instance, the best solution would see 10 elite ants march over it, plus 5 "ranked ants"; the second best solution would see zero elite ants and 4 "ranked ants" march over it; etc.

The pheromones, in their most basic form, represent information gleaned from past iterations that influences the current iteration, and are stored in a matrix $\tau = \{\tau_{ij}\}$ which covers all possible node combinations. After several iterations, the ant will prefer selecting the more optimum node connections due to their heavier weight in pheromone levels (Figure 5). By carefully selecting the weighting of the pheromone deposition levels and the rate of decay, near-optimum solutions can be teased from the random selections, as shown in Figure 6; note that the optimum path has a high pheromone weight and therefore a high probability of being selected, which further contributes to more pheromones being added to it in a virtuous cycle. Note also that there still exists a finite amount of pheromones in the alternate routes, leaving open the chance of exploring alternate routes.

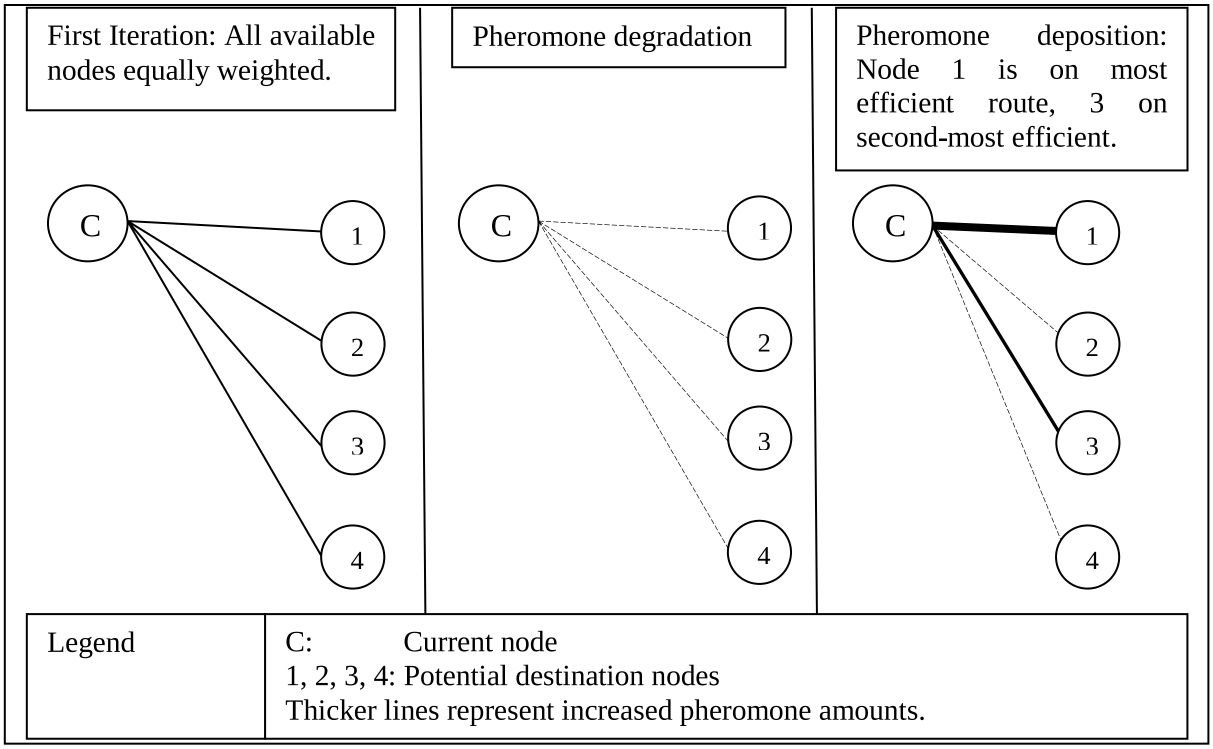


Figure 5. Effects of pheromone deposition

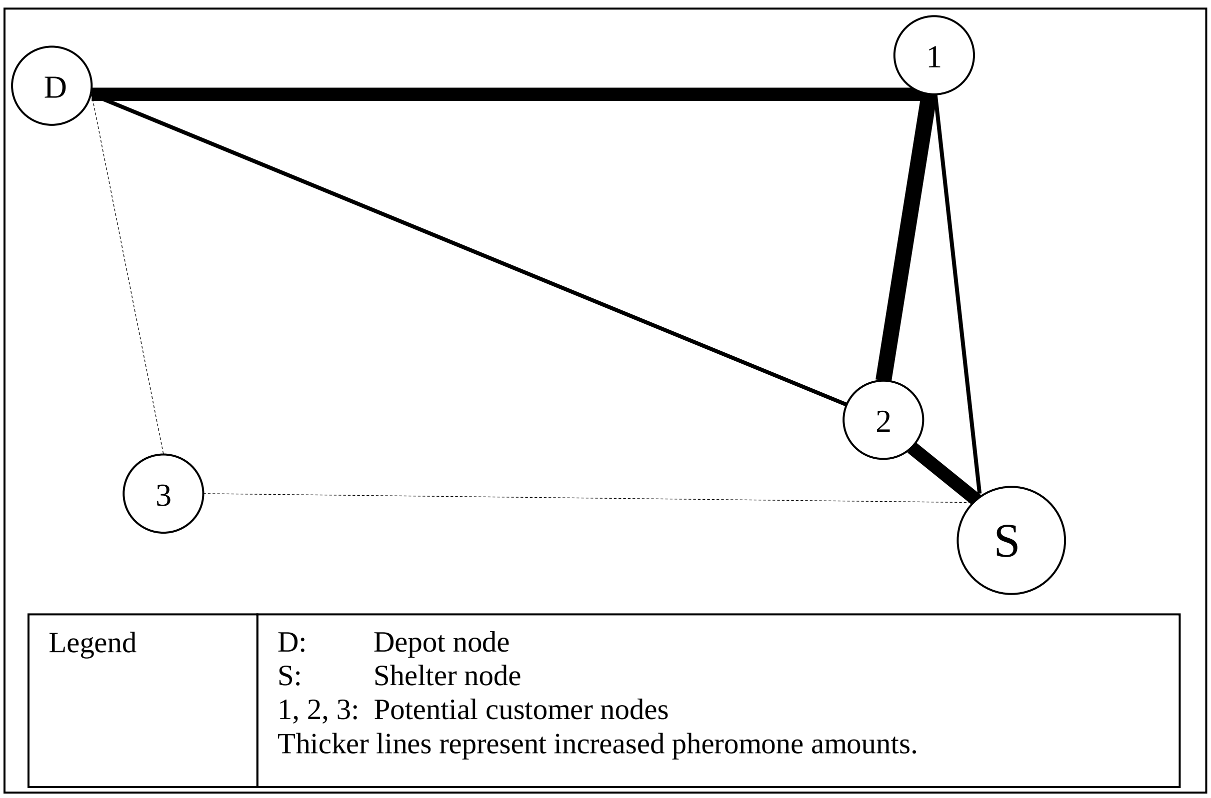


Figure 6. Example solution

2. Heuristic

In Dorigo and Stutzle's (2004) evaluation, the addition of a simple heuristic to the route construction procedures for the ants resulted in a convergence to an optimum with fewer iterations than a completely random approach. Dorigo, Maniezzo, & Colorni, (1996) compared this to the ant's "visibility." Two heuristics were chosen for this algorithm. The first, a neighborhood ant heuristic (η^1), entailed limiting the routes available to an ant to the nearest Ω nodes, with Ω being set by the user when the routine was called. The node weighting matrix for heuristic η^1 is defined in Equation 4.1.

$$(\textit{Neighborhood ant})\ \eta^1 := \left[\eta^1_{ij}\right] = \begin{cases} 1 & \text{if among nearest } \Omega \text{ nodes by travel time} \\ 0 & \text{otherwise} \end{cases} \forall(i, j) \quad (4.1)$$

The second ant heuristic, η^2, simply weighted the probability of the node being selected by the proximity of that node to the one that the ant currently sat on; near nodes were more likely to be selected than far nodes, although a finite probability of selection existed for far nodes at all times, in contrast with the first heuristic (Equation 4.2). This second type of ant draws parallels to the GRASP methodology, in terms of specifying a preference for a greedy route but leaving open the option for random perturbations from the greediest route. The GRASP method was described in II.C.3.a.(1).

$$(\textit{Greedy-random ant})\ \eta^2 := \left[\eta^2_{ij}\right] = \quad \text{travel time rank among available nodes}\ \forall(i, j) \quad (4.2)$$

To avoid limiting the routine to nodes specified by these two heuristics, a third type of ant was constructed, η^3, that relied exclusively on pheromones to guide its node selection (Equation 4.3). Figure 7 illustrates all three heuristics' philosophies.

$$(\textit{Pheromone-only ant})\ \eta^3 := \left[\eta^3_{ij}\right] = 1 \qquad \forall(i, j) \qquad (4.3)$$

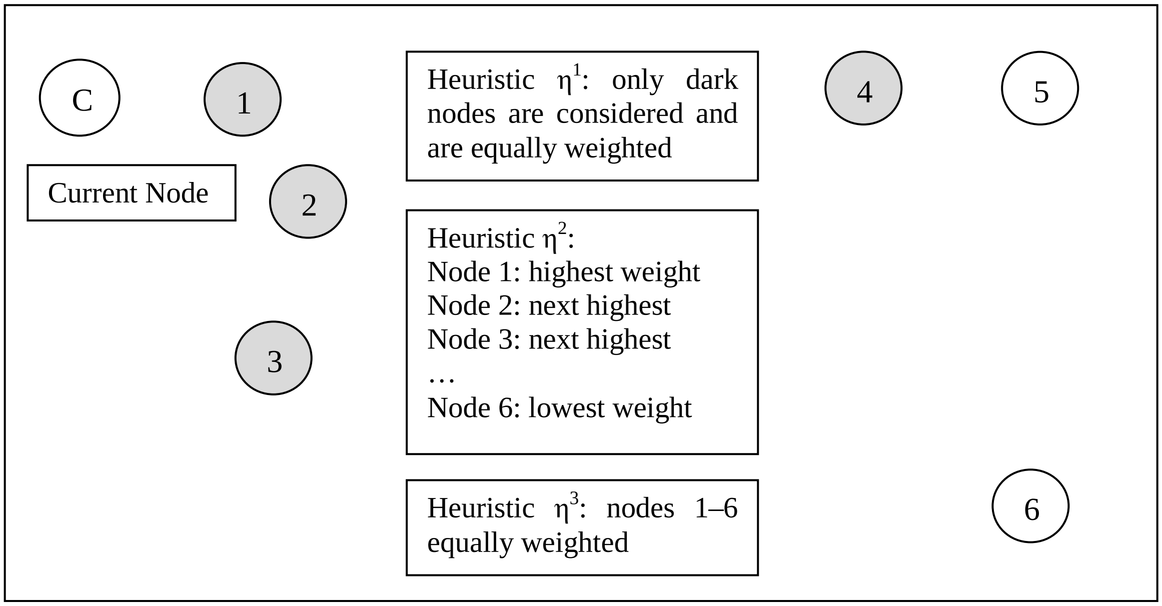


Figure 7. Heuristic node mapping

The algorithm executes the following steps. First, the ant type is selected at random between the nearest neighborhood ant (η^1), the greedy-random ant (η^2) or pheromone-only ant (η^3). The route construction starts at a randomly-chosen vehicle node. The list of un-selected nodes is filtered down to nodes that are able to fit travel time to the node, plus time to load the passenger, plus travel time to the shelter, plus time to unload the passenger, all without exceeding the total time available. This filtered list of nodes has weights η assigned according to the appropriate ant-heuristic, and then a second set of weights assigned according to the pheromone matrix τ. Each set of weights is proportioned exponentially according to values assigned when the routine is initially called: α for the pheromone weight and β for the heuristic weight.

τ and η are combined in two different fashions: one a normalized sum and the other a normalized product. During initial programming, the probabilities were normalized, weighted exponentially, then added to each other and renormalized. This approach was selected based on its simplicity, with the intention that the combination would be upgraded during a later revision to the formation presented by every other ACO in the literature. In the more standard formulation, the two weighted probabilities are multiplied by each other and then renormalized. The probability computation can be generalized for any node *i* to any node *j*, as shown in Equations 4.4 (additive formulation) and 4.5 (multiplicative formulation). As it turned out, the additive formulation generated relatively good results, so it was formally evaluated against the original formulation.

$$P_{ij} = \frac{\dfrac{\left[\tau_{ij}\right]^{\alpha}}{\sum_{n}\left[\tau_{in}\right]^{\alpha}} + \dfrac{\left[\eta_{ij}\right]^{\beta}}{\sum_{n}\left[\eta_{in}\right]^{\beta}}}{\sum_{n}\left(\dfrac{\left[\tau_{in}\right]^{\alpha}}{\sum_{n}\left[\tau_{in}\right]^{\alpha}} + \dfrac{\left[\eta_{in}\right]^{\beta}}{\sum_{n}\left[\eta_{in}\right]^{\beta}}\right)}, \ \forall\ n \text{ feasible nodes} \tag{4.4}$$

$$P_{ij} = \frac{[\tau_{ij}]^{\alpha} * [\eta_{ij}]^{\beta}}{\sum_{n}\left([\tau_{in}]^{\alpha} * [\eta_{in}]^{\beta}\right)}, \ \forall\ n \text{ feasible nodes} \tag{4.5}$$

The process of node selection continues node-by-node. When no nodes are available, the route is directed back to the shelter and the occupants are unloaded. If additional routes are allowed, the vehicle is sent back out, accumulating nodes as before. If additional routes are not allowed, a different vehicle is selected and the time is reset. If all vehicles have been used, a new ant is selected and the whole process is repeated.

Once all ants have been used – the number of ants is set by the user when the routine is called – the algorithm picks the best solutions, which minimize the cost function of the number of un-served customers. A user-specified number of best solutions are stored in a list.

The pheromone matrix τ is then degraded by multiplying by (1-ρ), where ρ is a user-specified parameter value less than one that represents the rate of decay of pheromone information. Next, for all node pair combinations T_{ij} which were traveled by a solution, pheromones Δτ are added to the pheromone matrix; the added pheromones are divided by the number of un-served customers *U* on the tour, which serves to dilute the effects of less-effective tours. Δτ is modified by a pair of multipliers. The first multipliers are the elite ants: a user-specified number σ of elite ants march over the best solution *B*.

The second multiplier depends on a user-specified binary value *Z*; if *Z* is 1, the pheromones are weighted by the solution's rank position in the best solution list *w*, with the most efficient solution receiving a higher weight than the less efficient solutions. When *Z* is 0, the option to weight pheromone addition by weight is turned off. The overall change in τ is encapsulated by Equation 4.6; note that even in the case of σ=0 and Z=0, pheromones Δτ are still added.

$$\tau_{ij}^{new} = \tau_{ij}^{old} * (1\text{-}\rho) + \sum_t \left(\frac{T_{ij}^t * \Delta\tau}{U_t} * (\sigma B + 1) * (\mathrm{Z} w^t + 1) \right), \text{ for all ants t}$$

where

$$T_{ij}^t = \begin{cases} 1 & \text{if edge } i, j \text{ was traversed by ant } t \text{ during tour} \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

$$B = \begin{cases} 1 & \text{for best solution} \\ 0 & \text{otherwise} \end{cases}$$

$$Z = \begin{cases} 1 & \text{for user option to use rank-based weighting} \\ 0 & \text{otherwise} \end{cases}$$

Once the pheromones have been updated, the procedure re-starts for another iteration, proceeding to a pre-identified iteration limit. In the case that time is not a limiting factor, such as for a small number of routes allowed per vehicle, an lower bound can be computed for the number of unserved customers, which would trip the procedure to conclude before the iteration limit is reached. For *N* customers, *K* vehicles with c_k capacity per vehicle, and *R* routes allowed, the lower bound of the number of un-served customers may be determined as shown in Equation 4.7.

$$S_{lowerbound} = \max(N - \sum_K c_k * R, 0) \tag{4.7}$$

To prevent the routine from becoming locked into a local optimum, a condition (Equation 4.8) was added that would reset the pheromone matrix if the best solutions remained the same for a number of iterations, using a procedure known as Pheromone Trail Smoothing (Stutzle & Hoos, 1997). The proportion of reset depends on the value of δ. The idea is to weaken the pheromone matrix, but not to completely abandon it; δ may range from 1 to 0, with δ=1 corresponding to a complete reset of τ to τ_{max}, and δ=0 corresponding to no reset at all.

$$\tau_{ij}^{new} = \delta\tau_{\max} + (1\text{-}\delta)\,\tau_{ij}^{old}, \text{ with } 0 < \delta < 1 \tag{4.8}$$

An overview of the pseudo-code is given in Figure 8. Table 1 gives a description of the full list of variables used in the algorithm.

Variable	**Description**
τ	Pheromone value
Ω	Number of nodes in neighborhood
α	Exponential weight to pheromones
β	Exponential weight to heuristic
σ	Number of elite ants
η	Heuristic value
ρ	Pheromone degradation multiplier
δ	Proportion of pheromone reset
$\Delta\tau$	Pheromone deposition amount
U	Number of un-served customers
w	Size of ranked candidate lists

Table 1. List of Variables Used in Algorithm

```
Loop iterations until iteration limit has been reached
        Loop Ants until all Ants have been used
                Select Ant (from neighborhood, greedy-random or pheromone only)
                Create Route
                Loop until all available nodes have been used
                        Select vehicle at random
                        Evaluate list of available nodes
                                Check for feasibility
                                Weight probabilities (Equation 4.4 or 4.5)
                        Select node
                        If all nodes have been used:
                                Send to shelter
                                reset vehicle
                                create additional routes if feasible (by sending back to beginning of loop)
                End route creation loop
        End Ant Loop
        Evaluate solutions
        Record best solutions
        Update pheromone matrix (Equation 4.6)
        If solutions are stuck in local optimum (i.e., best solutions list has not changed over several iterations)
                Reset pheromone matrix (Equation 4.8)
End iteration loop
```

Figure 8. Pseudocode for hybrid ant system algorithm

D. ALGORITHM EXAMPLE

To further illustrate how the routine works, a brief example follows.

1. Dataset and User Specified Values

To construct the dataset, the travel time between nodes was first established. This was done by placing the shelter at the origin and then randomly assigning integer coordinates to the remaining nodes using a routine. The number of customer nodes was set at 7 to allow a variety of solutions to be explored without becoming too complex. The model was laid out as shown in Table 2. Note that the time units are intentionally left without dimension, allowing the user to select whatever units are most desirable, as long as they remain consistent throughout the problem.

Variable	Value
Vehicles	1
Vehicle capacity	2
Number of disability levels	1
Number of allowed routes per vehicle	6
Time limit (nondimensional)	36
τ_{max}	1000
τ_{min}	.05
ρ	.2
Σ	0
$\Delta\tau$	500
α	2
β	2
Ω	4

Table 2. Parameter Values for Example Problem

The dataset is designed so that the time available is the limiting factor; the vehicles are not expected to have the time to perform six routes. Table 3 describes the remainder of the dataset. Figure 9 shows graphically the relative locations of the points in Euclidean travel-time space.

Node Description	Node Number	X-Coord	Y-Coord	Number of Customers	Load Time
Shelter	S	0	0		
Customer	1	1	6	1	6
Customer	2	5	8	1	2
Customer	3	5	6	1	1
Customer	4	-1	-1	1	2
Customer	5	-5	3	1	1
Customer	6	4	-6	1	1
Customer	7	1	-1	1	3
Depot	D	-2	-2		

Table 3. Dataset for Example Problem

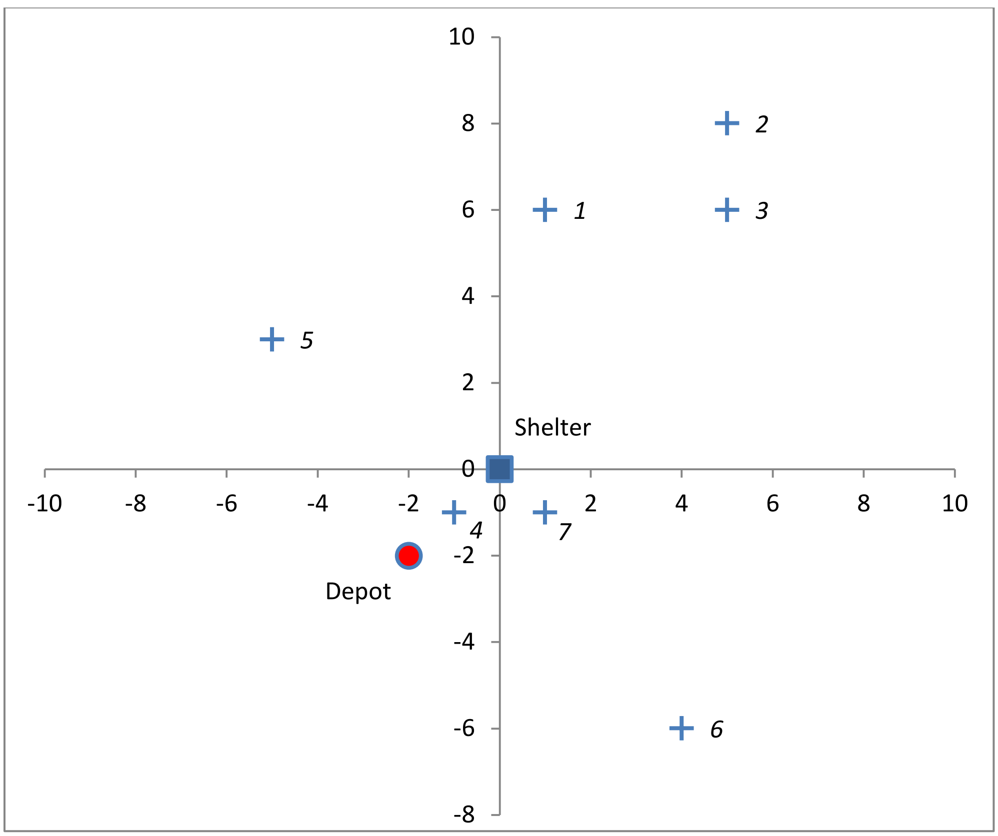


Figure 9. Location of Nodes in Travel-Time Space

2. Distance and Loadtime Matrix Calculation

In this example, we pre-calculate the travel time matrix, whereas the current version of the implemented algorithm only has partial pre-processing in place. We believe pre-processing is a good idea, as it limits the number of calculations and likely decreases the computation time. Table 4 shows the final distance and load time matrix; each row and each column represents a node. This will be decremented from the time remaining each time a node is selected. For this, and all other matrices discussed in this example section, the rows should be regarded as the “from” node and the columns as the “to” node. Note that once a vehicle has departe from the depot D, it will not revisit the depot, so there is consequently no D column.

		To							
		S	**1**	**2**	**3**	**4**	**5**	**6**	**7**
From	**S**	0	18.08	13.43	9.81	5.41	7.83	9.21	7.41
	1	6.08	12.00	8.47	6.00	11.28	8.71	14.37	13.00
	2	9.43	16.47	4.00	4.00	14.82	13.18	16.04	15.85
	3	7.81	16.00	6.00	2.00	13.22	12.44	14.04	14.06
	4	1.41	19.28	14.82	11.22	4.00	7.66	9.07	8.00
	5	5.83	18.71	15.18	12.44	9.66	2.00	14.73	13.21
	6	7.21	24.37	18.04	14.04	11.07	14.73	2.00	11.83
	7	1.41	19.00	13.85	10.06	6.00	9.21	7.83	6.00
	D	2.83	20.54	16.21	12.63	5.41	7.83	9.21	9.16

Table 4: Travel Time Plus Load Time Matrix

In Table 5 we construct a similar matrix to Table 4 but add the travel time from the customer node to the shelter; in this case adding the transposed column S to each row. This gives the minimum time required to serve a node’s customer, travel back to the shelter, and disembark the customer. It is distinct from distances in Table 4 because the return trip to the shelter changes with each customer node. The purpose of Table 5 is to filter the potential destinations from a node to only those nodes which can be served in the remaining time.

		To							
		1	2	3	4	5	6	7	D
From	S	24.17	22.87	17.62	6.828	13.66	16.42	8.828	5.657
	1	18.08	17.91	13.81	12.69	14.54	21.58	14.41	11.37
	2	22.55	13.43	11.81	16.23	19.01	23.25	17.26	15.03
	3	22.08	15.43	9.81	14.63	18.27	21.25	15.48	13.46
	4	25.36	24.25	19.03	5.414	13.49	16.28	9.414	4.243
	5	24.79	24.61	20.25	11.07	7.831	21.94	14.63	8.659
	6	30.45	27.47	21.85	12.49	20.56	9.211	13.25	10.04
	7	25.08	23.28	17.87	7.414	15.04	15.04	7.414	5.991
	D	26.63	25.64	20.44	6.828	13.66	16.42	10.58	2.828

Table 5. Minimum Time Left Required for each Node

3. First Iteration

Every element of the τ matrix is initialized to τ_{max}, as illustrated in Table 6.

		To							
		S	1	2	3	4	5	6	7
From	S	1000	1000	1000	1000	1000	1000	1000	1000
	1	1000	1000	1000	1000	1000	1000	1000	1000
	2	1000	1000	1000	1000	1000	1000	1000	1000
	3	1000	1000	1000	1000	1000	1000	1000	1000
	4	1000	1000	1000	1000	1000	1000	1000	1000
	5	1000	1000	1000	1000	1000	1000	1000	1000
	6	1000	1000	1000	1000	1000	1000	1000	1000
	7	1000	1000	1000	1000	1000	1000	1000	1000
	D	1000	1000	1000	1000	1000	1000	1000	1000

Table 6. τ Matrix Initialized at τ_{max}.

a. First Node

The first node of the first route will be the vehicle depot; if there were multiple vehicles, one depot would be randomly picked. In this case, node 8 is Depot D. Normally the type of heuristic is chosen for each ant and is kept through all nodes and routes for the full tour, but in this example we will illustrate all three heuristics. For the first node we choose the **random heuristic η^3**; that is, no heuristic weighting at all. The pheromone weighting is extracted from the τ matrix; in this case, it is trivial and all nodes are weighted with the value of τ_{max}, in this case 1000. We filter the nodes for time; the time available is the full time allotted, in this case 36. Since all elements of row D (the current node) of Table 5 are less than 36, all of the nodes are viable and are labeled with a binary "1" in Table 7; if any nodes had been non-viable, they would have been labeled with a "0" and their contribution would have been nullified when multiplied through. Table 7 enumerates the values assigned to each node. Next we filter for capacity; in this case, the capacity is trivial since there is only one disability level: the customer at each node will be able to fit into our vehicle. If there were multiple disability levels and the vehicle was only able to carry less-disable customers, as in the instance of a stretcher-bound customer and a vehicle unable to service stretchers, then those "stretcher nodes" would be filtered out. Absent the higher levels of disability, all nodes are given a "1" in Table 7.

The method of combining probabilities normally does not change during computation; for purposes of illustration, we change the combination method in this example. The probabilities of node selection for the second node will be computed using the **additive formulation**. First, each heuristic value is converted to a probability; that is, they are normalized so the sum is equal to 1; in this case, all the heuristic probabilities will be 1/(1+1+1+1+1+1+1) = 1/7 or .1429. Next the same process is applied to the pheromone values, with the same result of 1000/7000 or .1429. The next step in the algorithm raises each heuristic probability to the power of β and each pheromone probability to the power of α. The probabilities are then added together and multiplied by the binary operators from the time and capacity filters. The totals are re-normalized so their aggregate sum is 1. Table 7 shows the final probabilities for the choice of the second node. A random number generator selects the next node using a cumulative distribution based on the probability distribution; in this case, the random number is .054. When the random number is compared to the cumulative probability distribution for the nodes, the node chosen for the second node is node 1. The time is updated as follows: after traveling to this node, and reserving appropriate time to disembark the customers at the shelter node when the vehicle returns at the end of the route, the remaining time will be 15.46.

Node	Heuristic	Pheromone	Time	Capacity	Final
1	1	1000	1	1	0.142857143
2	1	1000	1	1	0.142857143
3	1	1000	1	1	0.142857143
4	1	1000	1	1	0.142857143
5	1	1000	1	1	0.142857143
6	1	1000	1	1	0.142857143
7	1	1000	1	1	0.142857143

Table 7. Values and Probability Assigned to Each Node from First Node

b. Second Node

Using the updated time left value of 15.46, the remaining nodes 2-7 are evaluated for feasibility based on row 1 from Table 5; the feasible nodes in this case will be 3, 4, 5, and 7. For this node we choose to illustrate the **greedy random heuristic η^2** shown in Equation (4.2); using Table 4, we rank the available nodes as 3, 5, 4, and 7 and assign values of 4, 3, 2, and 1 to each node, respectively. That is, node 3 ranks the highest because it has the lowest travel plus load time and is therefore assigned the highest rank of 4; when converted to a probability, node 3 would have a 40% probability of selection based on heuristics only, while node 7 would have only a 10% probability of selection.

Since this is still the first iteration, the τ matrix has not changed and all the pheromone weights will be 1000 again. Likewise, filtering each node for available vehicle capacity fails to eliminate any nodes. Table 8 shows the resultant probability distribution after a repeat of the additive combination process. With a random number generated by the computer of .033, the node chosen becomes node 3. The 6 time units computed for this node pair in Table 4 is subtracted from the previous time left of 15.46, leaving 9.46.

Node	Heuristic	Pheromone	Time	Capacity	Final
2	0	1000	0	1	0.00
3	4	1000	1	1	0.46
4	2	1000	1	1	0.16
5	3	1000	1	1	0.29
6	0	1000	0	1	0.00
7	1	1000	1	1	0.09

Table 8. Values and Probabilities Assigned to Each Node from Second Node

From the second node, a filter for capacity results in no feasible nodes, so the route is considered finished and the time left is decreased by the travel time to the shelter node, calculated in Table 4 as 7.81, to a value of 1.64. Thus, the nodes for the first route of the first tour are, in order: D, 1, 3, and S. Total travel time is 34.36 time units.

c. Second Route

The second route begins at the shelter node, node S. However, a glance at Table 5 reveals that the travel time required for each of the remaining nodes 2, 4, 5, 6, and 7 exceeds the time left of 1.64; therefore, ant 1 has completed its tour.

d. Pheromone Adjustment

If there were additional ants in this problem, each one would in turn create a tour. However, for simplicity, we used a single ant; therefore, the first iteration is complete after the single tour.

In between the iterations, the τ matrix is updated by Equation 4.6. The elements that receive pheromone deposits would be (D,1); (1,3); and (3,S). The updated τ matrix is shown in Table 9.

					To				
		S	1	2	3	4	5	6	7
	S	200	200	200	200	200	200	200	200
	1	200	200	200	300	200	200	200	200
	2	200	200	200	200	200	200	200	200
	3	300	200	200	200	200	200	200	200
From	4	200	200	200	200	200	200	200	200
	5	200	200	200	200	200	200	200	200
	6	200	200	200	200	200	200	200	200
	7	200	200	200	200	200	200	200	200
	D	200	300	200	200	200	200	200	200

Table 9. Updated τ Matrix.

4. Second Iteration

Much like the first iteration, the second iteration progresses route by route and node by node. The first node is the depot, node 8. Nodes 1–7 are feasible for capacity and for time. We use the **neighborhood heuristic** $\boldsymbol{\eta^1}$ here; since Ω=4, we use the four closest nodes, using row 8 of Table 4, which gives nodes 4, 5, 6, and 7. With the **multiplicative algorithm**, Equation 4.5, the probabilities resolve to those listed in Table 10. The random number generator gives .87 in this case, which means that node 7 is the next node.

Node	Heuristic	Pheromone	Time	Capacity	Final
1	0	300	1	1	0
2	0	200	1	1	0
3	0	200	1	1	0
4	1	200	1	1	0.25
5	1	200	1	1	0.25
6	1	200	1	1	0.25
7	1	200	1	1	0.25

Table 10. Values and Probabilities Assigned to Each Node from First Node, Second Iteration

5. 20th Iteration

To further illustrate the multiplicative algorithm, Equation 4.5, we fast-forward to the 20th iteration. By this iteration, the τ matrix has degraded in most places to its minimum value. For the sake of illustration, we continue the previous route from node 7, and remove the heuristic influence. The resulting probabilities are shown in Table 11.

Node	Heuristic	Pheromone	Time	Capacity	Final
1	1	0.05	1	1	2.40E-07
2	1	0.05	1	1	2.40E-07
3	1	0.05	1	1	2.40E-07
4	1	20	1	1	0.04
5	1	0.05	1	1	2.40E-07
6	1	100	1	1	0.96

Table 11. Values and Probabilities Assigned to Each Node from Second Node, 20th Iteration

6. Conclusion

The process continues until the number of iterations reaches a pre-defined limit.

An actual solution for this dataset is shown in Figure 10; the solution shows three un-served customers and two routes traveled by the single vehicle.

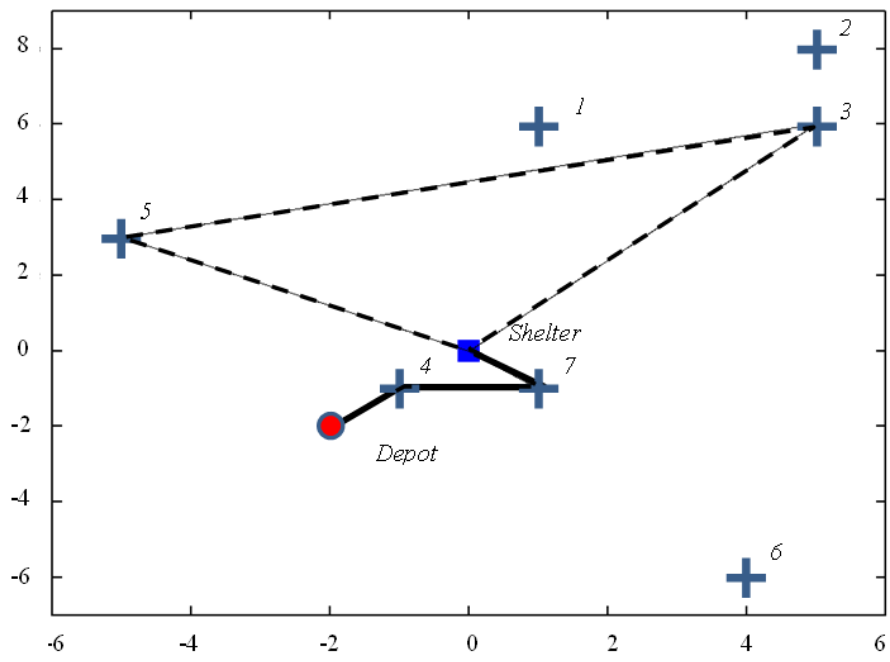


Figure 10. One Solution to Example Problem.

THIS PAGE INTENTIONALLY LEFT BLANK

V. EMPIRICAL ANALYSIS

This chapter presents the results from empirical investigations. Section A describes the test datasets used while Section B outlines the experimental set-up. Section C presents sample visual plots of the optimized route-tour while Section D compares the performance of the additive approach against Dorigo (2000)'s multiplicative approach. Section E identifies heuristic parameter settings that cater to a range of test scenarios.

A. TEST DATA

The complexity of the OB-VRP means that there is no readily-available standard datasets in literature to validate or benchmark the solution. As such, five new stylized datasets are constructed in this study to cover the diverse possible scenarios in terms of number of customers, disability level, vehicle capacities, total time available, etc. Their characteristics are summarized in Table 12. Detailed specifications are in Appendix A.

	Number of vehicles, K	Number of customers, N	Number of disability levels, L	Maximum number of tours allowed per vehicle, R	Total time available, T
Dataset 1	2	40	1	2	200
Dataset 2	4	40	3	5	500
Dataset 3	4	100	2	6	400
Dataset 4	2	15	1	4	15
Dataset 5	3	20	2	6	300

Table 12. Main Characteristics of Test Data

Datasets 1 and 4 represent simple scenarios with two vehicles and one disability level. Specifically, the node locations, load/unloading times and other parameters are prescribed in Dataset 4 such that the global optimal solution is known a priori, i.e., five un-served customers. Datasets 2 and 3 represent more complex cases with more vehicles, nodes and disability levels. Node locations and loading/unloading times for each dataset are randomly assigned (uniform distribution for Datasets 2, 3 and 5). Figure 11 illustrates the physical layouts of the five datasets.

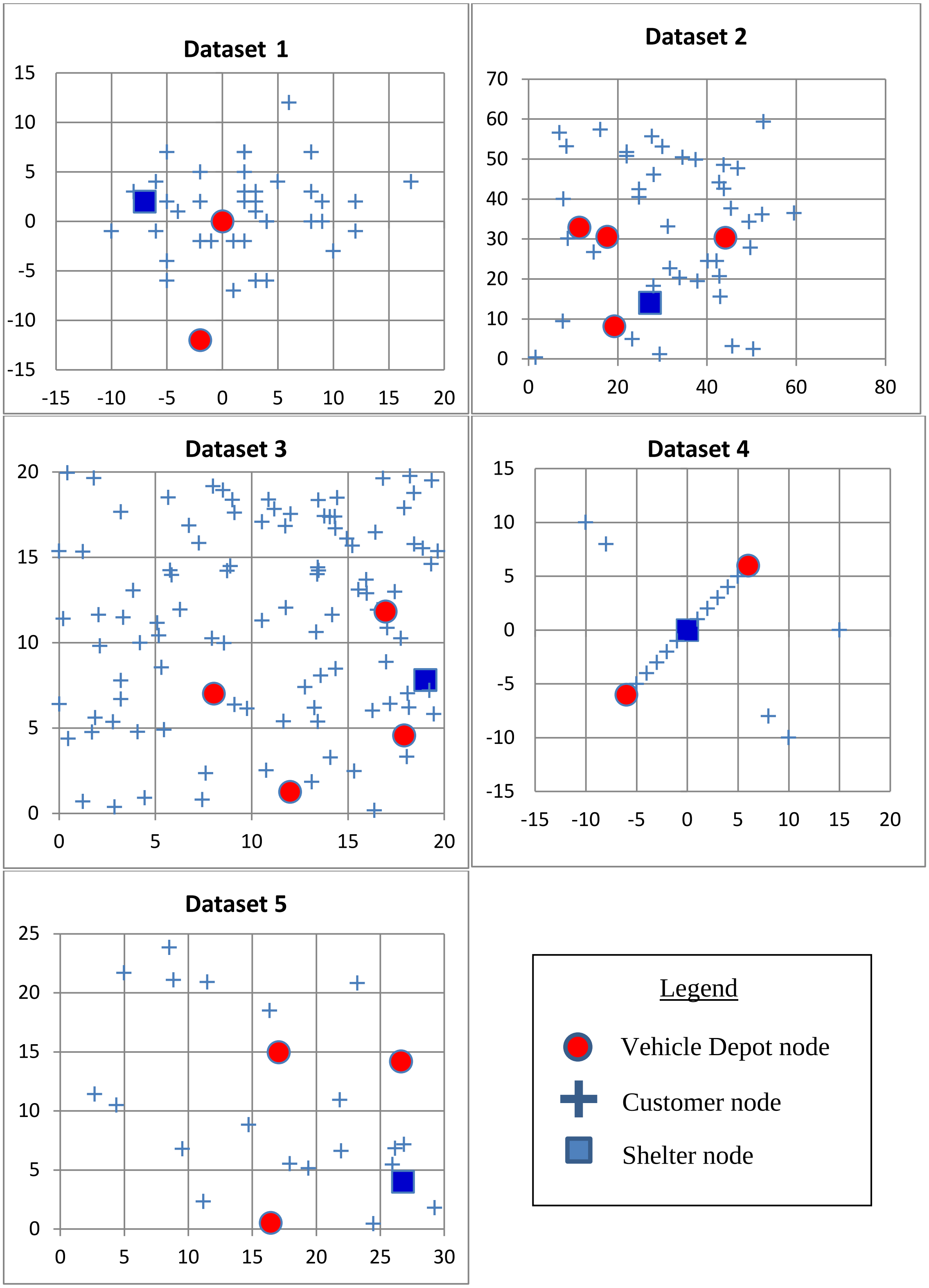


Figure 11. Layout of the five test datasets

The sizes of the datasets, ranging from 20 to 100 customer nodes, correspond to typical small to mid-size test instances found in VRP literature. The three disability levels used in Dataset 3 would represent real-world evacuees who are either stretcher-bound, wheelchair-bound or those who need walking aids.

B. TEST SET-UP

The hybrid metaheuristic was programmed using Octave 3.2.4 and run on two machines to reduce computation time: a desktop PC clone with an AMD Phenom II X3 720BE three-core processor at 3.0 GHz, 8 GB RAM, running Linux Ubuntu 10.04 64-bit, and a Lenovo PC laptop, with an Intel Centrino Core2 Duo CPU T5800 at 2.0 GHz, with 3GB RAM and running Linux Ubuntu 10.04 32-bit and Windows XP. The aims of the computational analysis are to determine (a) whether the additive or multiplicative algorithmic formulation performs better and (b) the choice of parameter value settings that would best suit a spectrum of test scenarios.

The computational analysis was executed using a Nearly Orthogonal Latin Hypercube (NOLH) design (Cioppa, 2002; Cioppa & Lucas, 2007; Kleijnen, Sanchez, Lucas & Cioppa, 2005). The NOLH experiment design allows the algorithm parameters to be varied over a wide range without requiring an exponential number of runs. For example, varying the parameter that governs the number of ants alone (Numants) at every integer level from 6 to 50 would require 45 scenarios. When combined with varying the parameter that governs the maximum pheromone level (τ_{max}) from 1 to 1000, this would require $45 \times 1{,}000 = 45{,}000$ scenarios. Further varying the other parameters would lead to a dramatic increase in the number of scenarios necessary. On the other hand, if each parameter is varied by using its lowest and highest values, there will only be $2^{11} = 2{,}048$ scenarios, but there will be no visibility of the parameter effects in the middle of their ranges.

The NOLH design can be used to select varying parameter values throughout their desired range in such a way that they have essentially no correlation with each other while keeping the number of scenarios low. With 11 factors, the NOLH design efficiently only requires 65 scenarios to obtain a broad representation of parameters in these ranges. Table 13 shows the input parameter ranges for the NOLH design[2] (the resultant parameter values for each of the 65 scenarios are found in Appendix B). The worksheet used to calculate the NOLH designs was developed by Sanchez (2005).

Parameter	m	$\Delta\tau$	ρ	τ_{max}	τ_{min}	α	β	Ω	σ	Candlist	Userrank
Lowest Value	6	0.01	0.001	1	0.01	0	0	3	0	1	0
Highest Value	50	10	0.999	1000	1	10	10	10	10	6	1
Decimal stepwidth	0	1	3	0	2	1	1	0	0	0	0

Table 13. Parameter Ranges for NOLH Experiment Designs (Number of Iterations = 40)

Variability in the analysis output comes from two sources. The first variability is the different demand levels from the five datasets that form the input. The second is the seed value used for the random number generator within the algorithm. With 65 scenarios for each of the five datasets and five random seed values per dataset, 65 x 5 x 5 = 1,625 replications would be generated. This variation in output is harnessed for the purpose of computational analysis in terms of comparing the performance of the additive and multiplicative algorithms as well as finding good parameter settings for the chosen algorithm.

C. PRELIMINARY RESULTS

Figure 12 presents a sample visual demonstration of the optimized route-tour schedule computed for Dataset 5.

[2] Another NOLH design version with more ants (70 to 200 ants, versus 6 to 50) was also developed and tested in our experimentattion. However, the significantly longer overall computational time (due to the 1,625 replication runs) led to the adoption of the fewer-ant version in order to speed up empirical analyses.

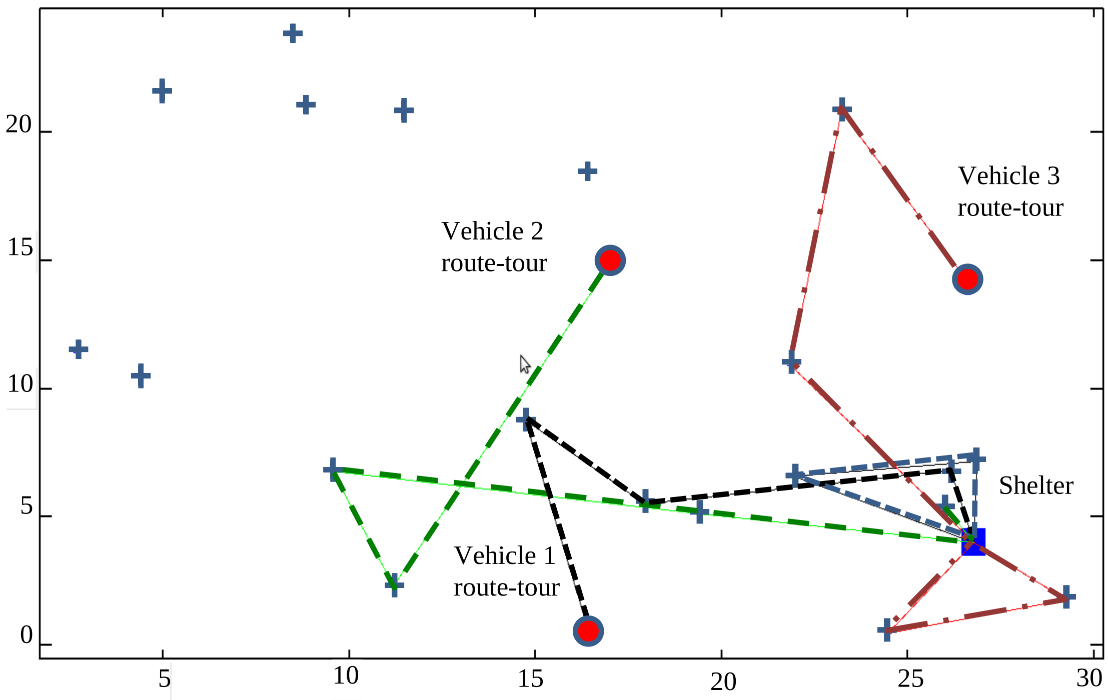


Figure 12. Graphical plots of route-tour solution for Dataset 5 (additive algorithm)

The plots of the OB-VRP route-tours are reminiscent of diagrammatic solutions for the traditional TSP or VRP. Nonetheless, the route-tours are not always as neat, with tendency to criss-cross, e.g., Vehicle 2. This is a likely result of the constraints imposed, especially the disability level of customers, that restricts the type and route of vehicles that can serve it. For criss-crosses within the same trip, it may also be due to suboptimal route construction. This can be mitigated with more convergence iterations, or introducing 2-opt and 3-opt local searches and/or a MACS element in the algorithm.

D. COMPARISON OF ADDITIVE VS MULTIPLICATIVE ALGORITHMS

Figure 13 and Figure 14 show respective differences in average solution quality (number of un-served customers in optimum solution) and average convergence time (number of iterations to converge) between the additive and multiplicative algorithms. The former performance metric is critical as it directly pertains to number of potential lives lost. The latter, speed, is of somewhat less importance as even the largest and most complex 100-node, 3-disability level, 4-vehicle dataset only takes about three minutes to converge. In a real-world scenario, the number of customer nodes in a city may scale into the tens of thousands such that algorithmic speed then plays a bigger role as a performance metric. Nonetheless, it is always possible to keep computational times within reasonable levels by harnessing parallel computing during implementation.

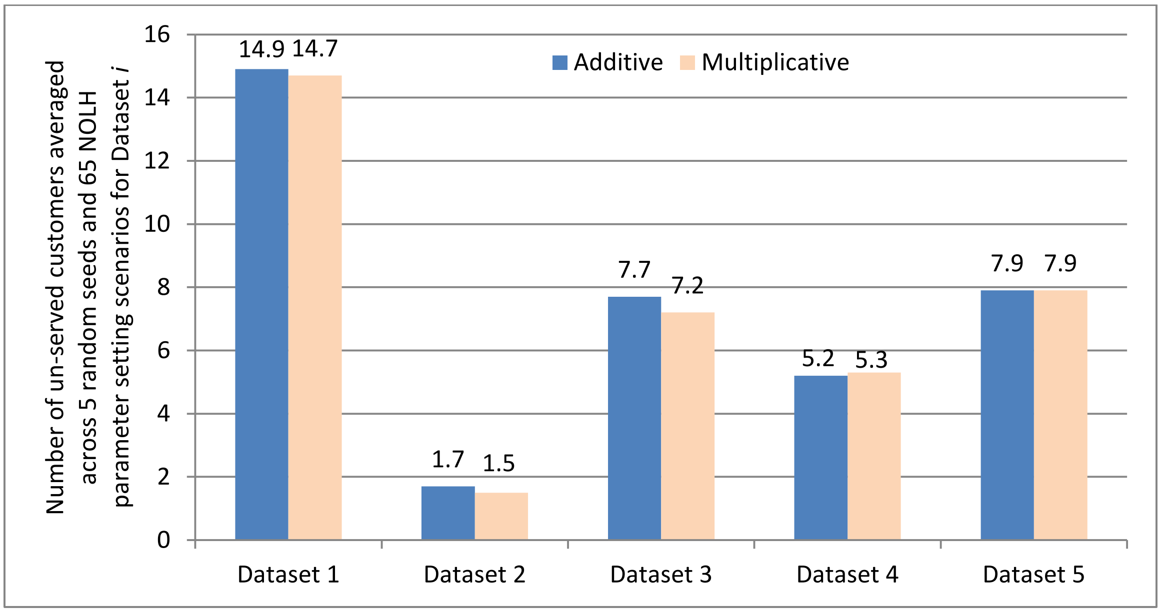


Figure 13. Comparison of average best solution (lower is better).

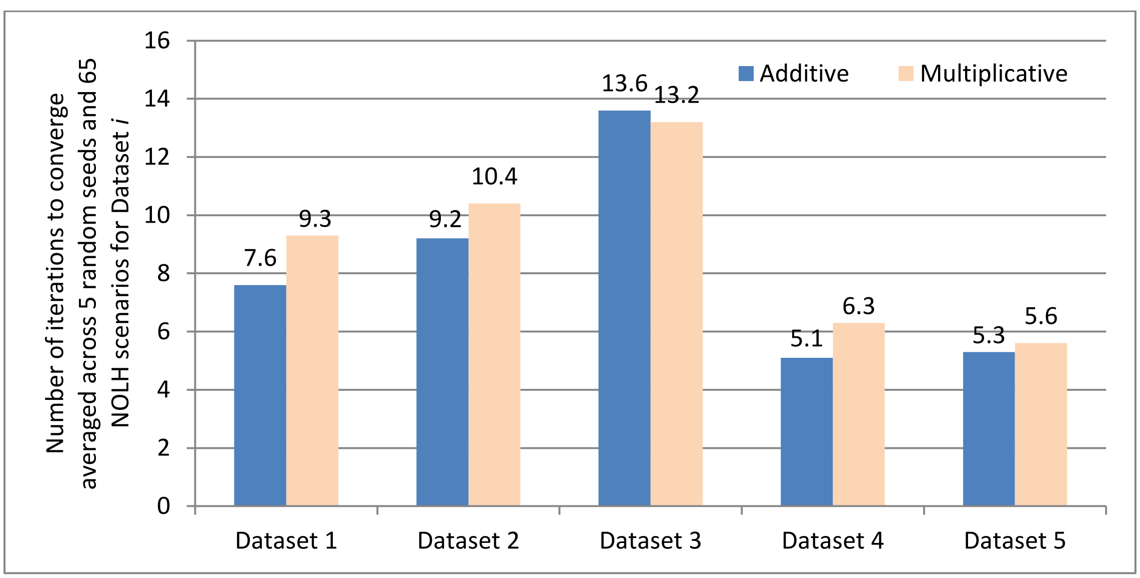


Figure 14. Comparison of average convergence times (lower is better).

Overall, the additive algorithm achieves comparable solution quality, typically more quickly. Critically, when the global optimum is known (five un-served customers in Dataset 4), the additive algorithm outperforms the multiplicative formulation.

To further isolate the effect of algorithm choice, a second set of comparison test was conducted where the heuristic element was excluded,setting $\beta = 0$ such that the algorithms ran with only the pheromone component. Results are in Figure 15 and Figure 16.

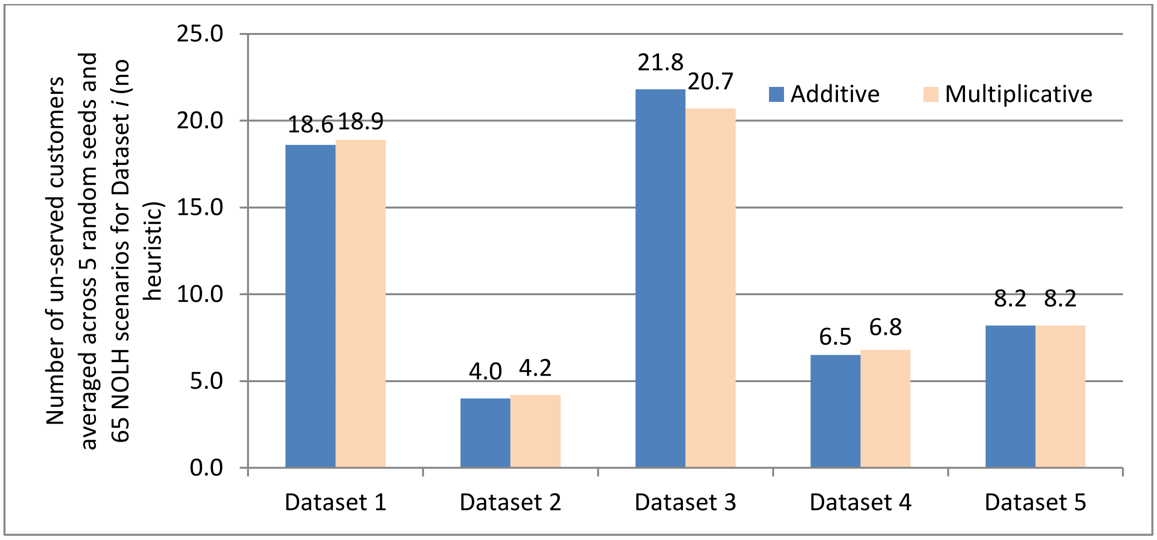


Figure 15. Comparison of average best solution, excluding heuristic component.

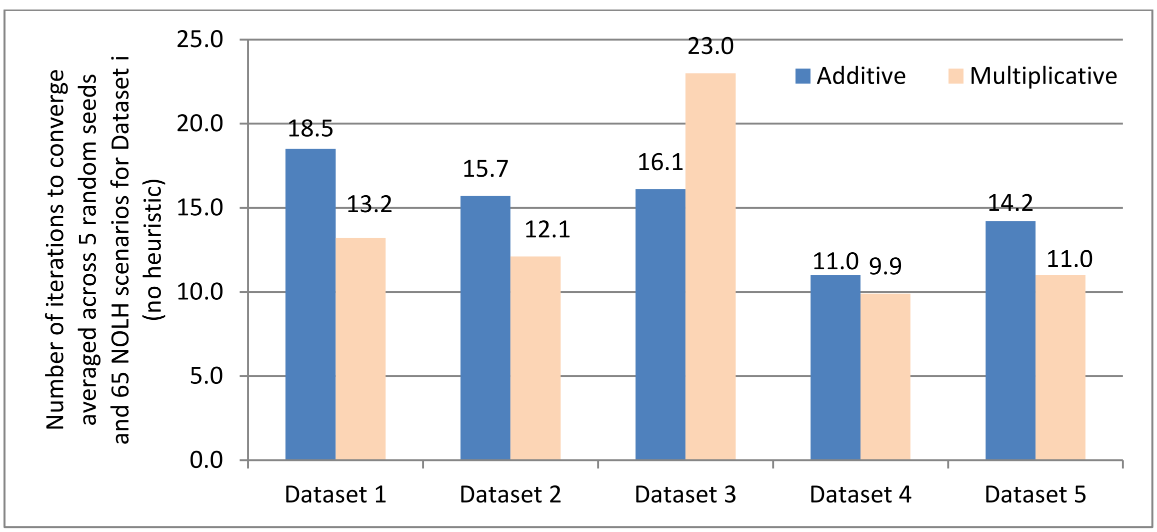


Figure 16. Comparison of average convergence times, excluding heuristic component.

Under the no-heuristic configuration, the additive algorithm generally produces better solution quality, albeit taking slightly longer convergence time (with the exception of Dataset 3). In particular, for Dataset 4 where the global optimum is known, the additive algorithm generally produced better solutions under both heuristic and no-heuristic configurations. Nonetheless, the additive algorithm did perform worse in the larger, most realistic scenario in Dataset 3.

While both formulations invoke the concept of averages in computing probabilities, the additive formulation calculates the arithmetic mean while the multiplicative formulation uses the geometric mean. The different results can be explained by examining the dissimilar probability distributions of choosing the next node, (for a heuristic ant sitting on a given node) between the two formulations. Table 14 illustrates how the additive formulation more widely expands the search space and evenly distributes the probabilities of choosing the next node, compared to the multiplicative formulation. For example in the case of the neighborhood ant when pheromone level τ is very high at 500, the additive formulation moderates the probability of choosing that node from 99.8% (in the multiplicative formulation) to a more restrained 75.7%, thus imparting a non-zero chance for other nodes to be considered candidates, even though their pheromone levels may be magnitudes lower. The same phenomenon is repeated for the GRASP-like heuristic. In conjunction with the longer search period that the additive formulation takes, this has allowed it to find a better solution generally compared to the multiplicative formulation.

In view of the overall better performance proffered by the additive formulation, subsequent empirical analysis will focus on the additive algorithm.

Customer node	τ (Pheromone level)	Probability of node being chosen for η^1 (neighborhood ant)		Probability of node being chosen for η^2 (GRASP ant)	
		Additive	Multiplicative	Additive	Multiplicative
1	12	0.0563	0.0006	0.0297	0.0008
2	10	0.0562	0.0004	0.0255	0.0005
3	500	0.7565	0.9988	0.8465	0.9947
4	2	0.0559	0.0000	0.0181	0.0000
5	8	0.0561	0.0003	0.0151	0.0002
6	6	0.0001	0	0.0122	0.0001
7	13	0.0005	0	0.0101	0.0003
8	9	0.0002	0	0.0076	0.0001
9	10	0.0003	0	0.0057	0.0001
10	60	0.0101	0	0.0156	0.0025
11	18	0.0009	0	0.0035	0.0001
12	50	0.0070	0	0.0096	0.0006
13	4	0.0000	0	0.0006	0.0000
14	4	0.0000	0	0.0002	0.0000
	Total	1	1	1	1

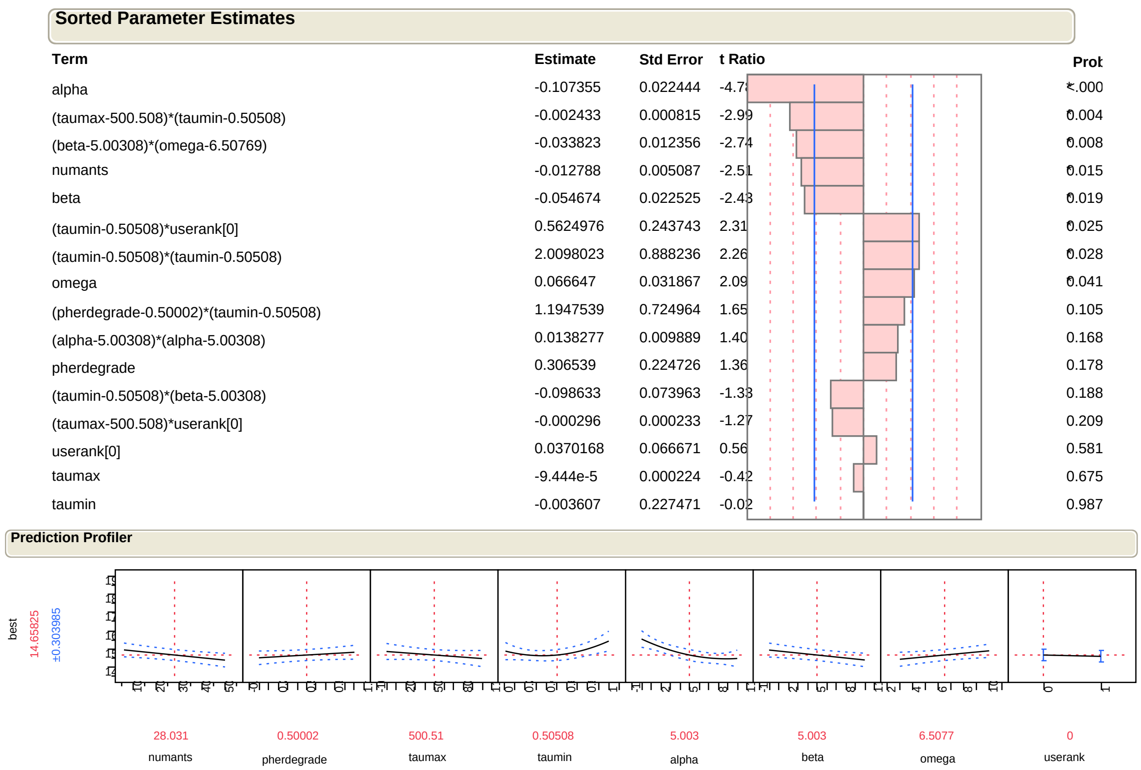
Table 14. Comparison of Probability Distribution of Choosing the Next Node Between Additive and Multiplicative Algorithms ($\alpha = 2$, $\beta = 2$. $\Omega = 5$ for η^1. τ values are random).

E. PARAMETER EFFECTS

The next step is to identify a suitable set of parameter value settings for the additive algorithm that best caters to a wide range of dataset instances. The analysis focuses on establishing appropriate parameter settings that produce better solution quality rather than relatively minor improvements in convergence times.

The JMP software (Version 9.0.1) is used to analyze the computational results. A regression model is fitted to the data using stepwise regression. The dependent variable is the number of un-served customers, while the covariates (independent variables) are the parameters, as well as their 2nd-order polynomial terms and factorial interaction terms. The stepwise regression stopping rule adopts a p-value threshold of 0.25 and employs bi-directional steps. The sorted parameter estimates and the prediction profiler charts are then used to ascertain the general behavior of the solution quality (number of un-served customers) as each parameter is varied.

JMP results for the average best solutions for Datasets 1 to 5 are shown in Figure 17 to Figure 21, respectively (see Appendix C for full JMP outputs). The estimated coefficient for each parameter represents its partial effect on the number of un-served customers, i.e., on average (across 65 NOLH scenarios and 5 random seeds), the increase (or decrease) in number of un-served customers for every unit increase in the parameter value, holding all other parameters constant.



Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob
alpha	-0.107355	0.022444	-4.7	<.000
(taumax-500.508)*(taumin-0.50508)	-0.002433	0.000815	-2.99	0.004
(beta-5.00308)*(omega-6.50769)	-0.033823	0.012356	-2.74	0.008
numants	-0.012788	0.005087	-2.51	0.015
beta	-0.054674	0.022525	-2.43	0.019
(taumin-0.50508)*userank[0]	0.5624976	0.243743	2.31	0.025
(taumin-0.50508)*(taumin-0.50508)	2.0098023	0.888236	2.26	0.028
omega	0.066647	0.031867	2.09	0.041
(pherdegrade-0.50002)*(taumin-0.50508)	1.1947539	0.724964	1.65	0.105
(alpha-5.00308)*(alpha-5.00308)	0.0138277	0.009889	1.40	0.168
pherdegrade	0.306539	0.224726	1.36	0.178
(taumin-0.50508)*(beta-5.00308)	-0.098633	0.073963	-1.33	0.188
(taumax-500.508)*userank[0]	-0.000296	0.000233	-1.27	0.209
userank[0]	0.0370168	0.066671	0.56	0.581
taumax	-9.444e-5	0.000224	-0.42	0.675
taumin	-0.003607	0.227471	-0.02	0.987

Figure 17. Dataset 1: JMP output for average best solution

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob
beta	-0.172617	0.011958	-14.	<.000
alpha	-0.099588	0.011958	-8.33	<.000
numants	-0.013456	0.002714	-4.96	<.000
(numants-28.0308)*(pherdegrade-0.50002)	-0.05945	0.014347	-4.14	0.000
(pherdegrade-0.50002)*(pherdegrade-0.50002)	1.7107513	0.478801	3.57	0.000
(alpha-5.00308)*(alpha-5.00308)	0.0188542	0.005693	3.31	0.001
(alpha-5.00308)*(beta-5.00308)	0.0162713	0.005952	2.73	0.008
(taumax-500.508)*(taumin-0.50508)	-0.001106	0.000431	-2.56	0.013
(taumin-0.50508)*(alpha-5.00308)	0.1005417	0.041658	2.41	0.019
(taumax-500.508)*(alpha-5.00308)	-7.118e-5	0.000043	-1.65	0.104
(taumax-500.508)*(beta-5.00308)	-8.724e-5	6.529e-5	-1.34	0.187
pherdegrade	0.0621312	0.11975	0.52	0.606
taumin	-0.047386	0.120753	-0.39	0.696
taumax	3.9189e-6	0.00012	0.03	0.974

Prediction Profiler

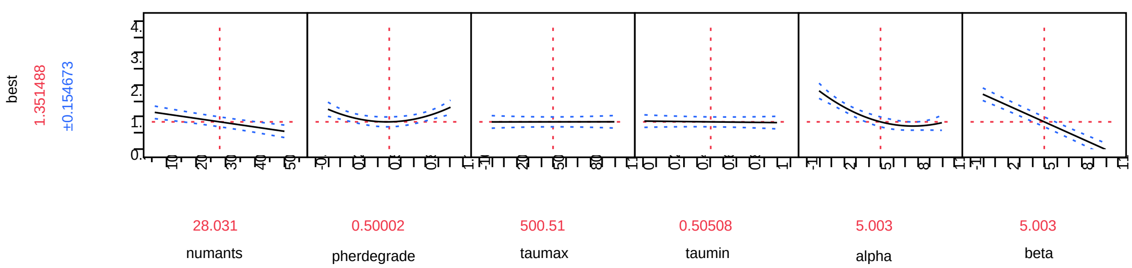


Figure 18. Dataset 2: JMP output for average best solution

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Pro
beta	-0.466305	0.081305	-5.7	<.00
alpha	-0.306964	0.081022	-3.79	0.00
omega	0.4264673	0.114719	3.72	0.00
(beta-5.00308)*(omega-6.50769)	-0.152811	0.045246	-3.38	0.00
(alpha-5.00308)*(alpha-5.00308)	0.1074928	0.032584	3.30	0.00
(taumin-0.50508)*(taumin-0.50508)	8.0191182	3.274625	2.45	0.01
(taumax-500.508)*(taumin-0.50508)	-0.006765	0.002902	-2.33	0.02
(alpha-5.00308)*(beta-5.00308)	0.0510794	0.033095	1.54	0.12
taumin	0.4166363	0.818197	0.51	0.61
taumax	0.00014	0.000811	0.17	0.86

Prediction Profiler

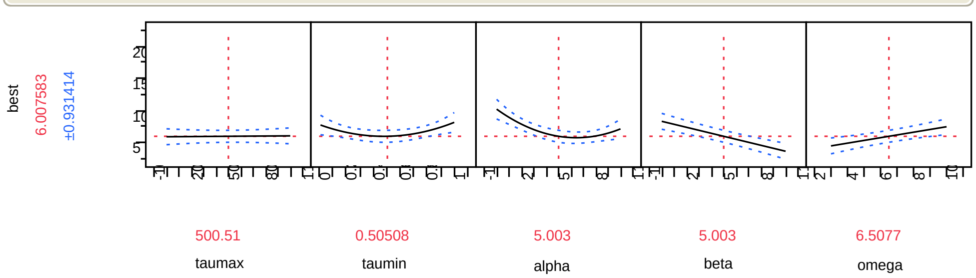


Figure 19. Dataset 3: JMP output for average best solution

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob
beta	-0.084351	0.008663	-9.74	<.000
(alpha-5.00308)*(alpha-5.00308)	0.0175901	0.004219	4.17	0.000
alpha	-0.03386	0.008654	-3.91	0.000
(alpha-5.00308)*userank[0]	0.036172	0.009836	3.68	0.000
(alpha-5.00308)*(beta-5.00308)	0.0162017	0.00461	3.51	0.001
(pherdep-5.00462)*(beta-5.00308)	0.0117169	0.003741	3.13	0.003
numants	-0.005873	0.001961	-2.99	0.004
(Candlist-3.50769)*userank[0]	0.060065	0.021652	2.77	0.008
(beta-5.00308)*(beta-5.00308)	0.0122354	0.004676	2.62	0.012
(numants-28.0308)*(taumin-0.50508)	0.0249629	0.009642	2.59	0.013
(pherdep-5.00462)*(Candlist-3.50769)	-0.012424	0.006285	-1.98	0.054
pherdegrade	0.1698374	0.086649	1.96	0.056
Candlist	0.0309212	0.016594	1.86	0.069
taumin	-0.160588	0.087693	-1.83	0.074
(taumax-500.508)*(taumin-0.50508)	-0.000577	0.000326	-1.77	0.083
(taumin-0.50508)*(alpha-5.00308)	0.0502044	0.030352	1.65	0.105
(numants-28.0308)*(pherdegrade-0.50002)	-0.01936	0.01284	-1.51	0.139
(taumax-500.508)*userank[0]	0.0001536	0.000109	1.41	0.165
pherdep	-0.011821	0.008646	-1.37	0.178
(taumax-500.508)*(taumax-500.508)	-5.77e-7	4.584e-7	-1.26	0.215
userank[0]	-0.026954	0.025607	-1.05	0.298
taumax	5.1783e-5	8.658e-5	0.60	0.553

Prediction Profiler

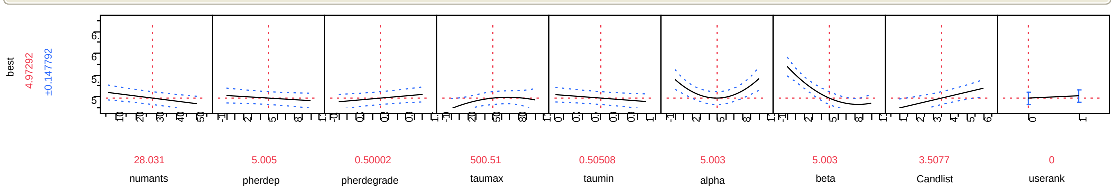


Figure 20. Dataset 4: JMP output for average best solution

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob
numants	-0.004835	0.000854	-5.6	<.000
alpha	-0.015694	0.003766	-4.17	0.000
(pherdep-5.00462)*(Candlist-3.50769)	-0.009284	0.002319	-4.00	0.000
(pherdegrade-0.50002)*(omega-6.50769)	-0.073264	0.023269	-3.15	0.002
(numants-28.0308)*(Candlist-3.50769)	0.0017243	0.000608	2.84	0.006
(alpha-5.00308)*(alpha-5.00308)	0.0046814	0.001692	2.77	0.008
pherdegrade	0.0931283	0.037714	2.47	0.017
(beta-5.00308)*(omega-6.50769)	0.0049153	0.002103	2.34	0.023
pherdep	0.0083437	0.003762	2.22	0.031
Candlist	0.0156549	0.00722	2.17	0.035
(alpha-5.00308)*userank[0]	0.0089431	0.004424	2.02	0.048
userank[0]	0.021938	0.01111	1.97	0.054
(beta-5.00308)*(Candlist-3.50769)	-0.004782	0.002656	-1.80	0.078
(numants-28.0308)*(pherdep-5.00462)	0.000463	0.000288	1.61	0.114
beta	-0.005824	0.003784	-1.54	0.130
(numants-28.0308)*userank[0]	0.0011725	0.00088	1.33	0.189
omega	0.0041357	0.005348	0.77	0.443

Prediction Profiler

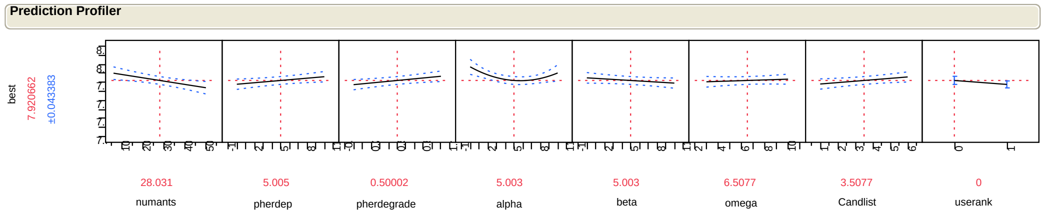


Figure 21. Dataset 5: JMP output for average best solution

Results and discussion of the empirical results are as follows.

1. Comparative Importance of Parameters

The relative significance of parameters varies considerably across datasets.

Some parameters are consistently important. The pheromone weight (α), heuristic weight (β) and number of ants used (Numants), as well as their 2nd order and interaction terms, repeatedly rank among the topmost critical parameters for all datasets (see Tornado Charts in Figure 17 to Figure 21).

Importance of other parameters varies. The pheromone level limits (τ_{min} and τ_{max}) are somewhat significant for most datasets (Datasets 1, 2 and possibly 3 and 4, covering a fairly broad range of scenarios (15 to 100 nodes, two to four vehicles and one to three disability levels), albeit less so than α, β and number of ants. Nonetheless, they are irrelevant for Dataset 5. On the other hand, pheromone deposit ($\Delta\tau$) and pheromone degradation (ρ) figure prominently for Datasets 2, 4 and 5, but play a less important role for Dataset 1 and 3.

The distinction is especially marked for the size of the neighborhood heuristic (Ω), use of ranked ants (Userrank) and the length of the best candidate list (Candlist). For example, although Ω has some effect for Datasets 1, 3 and 5, it is inconsequential for Datasets 2 and 4. Similary, while the size of Candlist and Userrank are important for Dataset 4 and 5, they have little effect for Datasets 2 and 3.

Some parameters are consistently insignificant. The number of elite ants used (σ), is found to have minimal influence on solution quality for all five datasets.

2. Direction of Parameter Effects

In a similar vein, the direction of the relationships between parameters and solution quality varies for different parameters.

The pheromone weight (α), heuristic weight (β) and number of ants used (Numants) exhibits consistent effects on the number of un-served customers. These three parameters largely display decremental effects across all datasets, i.e., a higher pheromone or heuristic weight or having more ants reduces the number of un-served customers. Where the stepwise regressions show significant influence, pheromone degradation (ρ), size of candidate list (Candlist) and size of neighborhood heuristic (Ω) show weak incremental effects. Detailed discussion of these results follow.

Number of ants. The observed effect is similar to that found in literature, although the degree varies. The greater the number of ants (i.e., a large colony), the greater exploration of possible routes-tours and hence the better the solution proffered. Using too few ants may result in a breakdown of their cooperative behavior due to the reduced communication and quick evaporation of pheromones along the trails. Nonetheless, having too many ants slows down computation time considerably. Dorigo and Stutzle (2004) suggest setting the number of ants to be the number of customer nodes. The results support this recommendation in terms of balancing solution quality verus efficiency.

Pheromone and heuristic weights. Figure 22 illustrates the effect of pheromone and heuristic weights on probability of choosing a particular node. Stronger pheromone weights (α), i.e., making the ants more sensitive to pheromones, whereas low pheromone weights would tend towards the classical stochastic greedy algorithm. Similarly, if heuristic weight (β) is set near zero, only pheromone is used without any heuristic bias. This generally leads to poor results with the rapid emergence of a stagnation situation, i.e., all the ants follow the same path and construct the same suboptimal route-tour (Dorigo, 1992; Dorigo et al., 1996). To provide good optimization dynamics, Dorigo et al. (1996) recommends setting $\beta \geq \alpha$. Dorigo and Stutzle (2004) further recommends setting α to be around one and β to be in the range of two to five based on experimental study of various ACO algorithms. This is generally in line with our recommended setting of 0.5–2 for α and 1–4 for β.

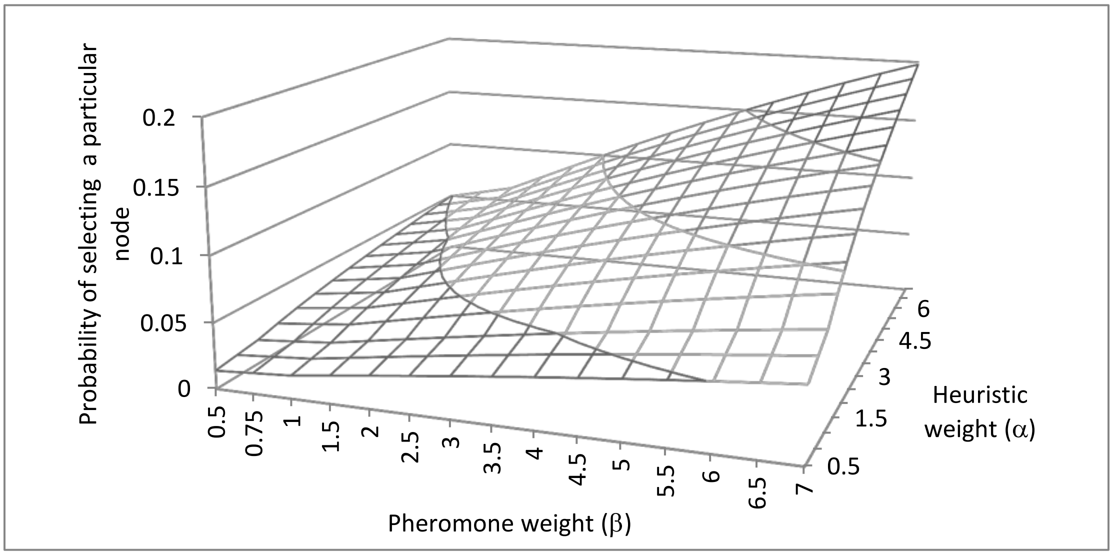


Figure 22. Effect of heuristic and pheromone weights (α and β) on probability of choosing a particular node.

Pheromone degradation. Figure 23 demonstrates the theoretical effect of pheromone degradation on the probability of choosing a particular node. Stronger degradation (ρ = 0.9) leads to higher evaporation such that ants have a "shorter-term" memory and recent search results are emphasized compared to older ones. This causes convergence in fewer (five) iterations during the search phase but suffers from a higher probability of choosing the (wrong) next node during the optimization phase due to excessive growth of pheromones on arcs of a suboptimal tour. Weaker degradation (ρ = 0.5) take a longer time to converge (15 iterations) during the search phase but gives a more diversified distribution of probabilities to nodes, thus allowing more thorough exploration at the start of route construction, and ultimately better overall better solutions during the optimization phase. The empirical results seem to indicate the reverse effect, but this can be attributed to the experimental constraint of 40 iterations. Given sufficient iterations, the prolonged search phase and more improved asymptopic behavior during the optimization phase should prevail and produce a better overall solution. In addition, since the pheromone matrix is initialized at τ_{max}, therefore in order to achieve convergence, the non-optimal nodes will have to degrade to a minimal value, while the more optimal nodes will stand out from the continual deposits. The key is thus to select an appropriate pheromone degradation rate that is not too high to avoid the situation where the ant colony prematurely forgets its past experience gained (i.e., loss of collective memory), hence impeding the ants' cooperative behavior. Dorigo and Stutzle (2004) recommend setting phereome evaporation to be in the range of 0.02 to 0.5. Based on our results, we recommend setting it at 0.1.

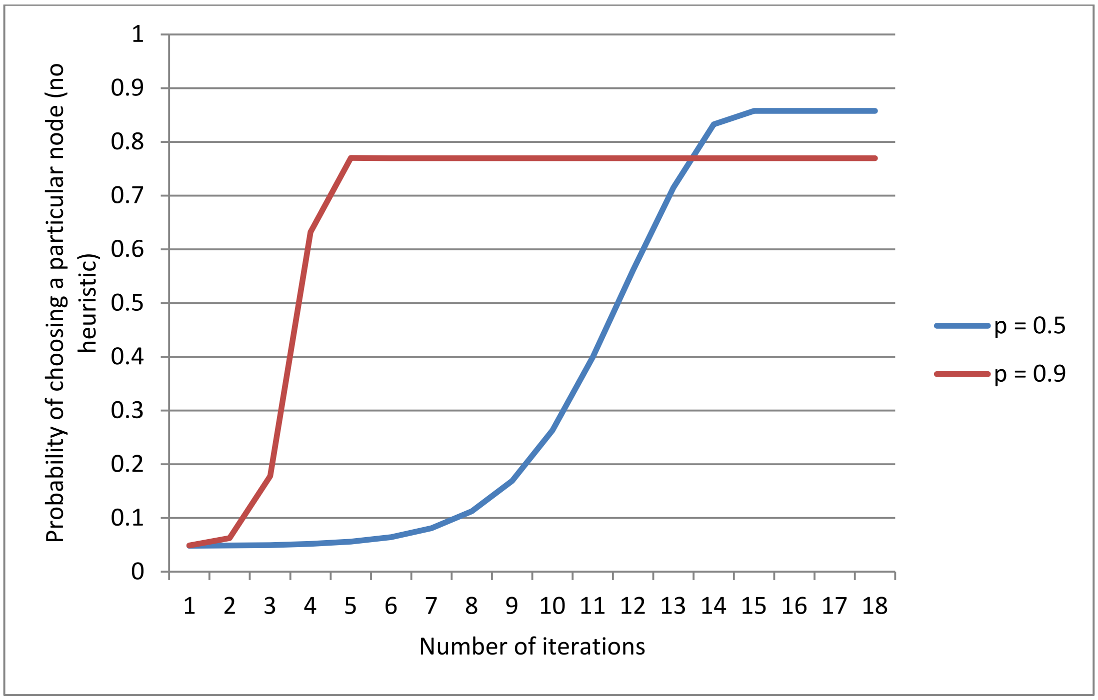


Figure 23. Effect of pheromone degradation (ρ) on probability of choosing a node ($\tau_{max} = 1000$, $\tau_{min} = 0.05$)

Size of neighborhood heuristic. A smaller neighborhood heuristic (Ω) would make the heuristic greedier in the sense that it would prevent the selection of nodes further away. If the heuristic is set too large (i.e., too many neighbor nodes are included), the ant becomes less heuristic in behavior such that the best solution produced will be constrained by the overall number of iterations (40) in the experiments. Therefore, given sufficient iterations, the results should even out such that a smaller neighborhood heuristic yield better overall solutions. Put differently, when the neighborhood size is equal to the total number of nodes, there is no heuristic influence at all because no nodes are eliminated from consideration. We recommend setting Ω between 3 to 8.

Length of Candidate list. A longer candidate list renders the algorithm less like a MMAS and more like a traditional ACS. Nonetheless, while a longer list allows more exploration during the search phase, a shorter list serves the optimization phases better as, the longer the list, the more poor-quality solutions are considered. If the problem is small to medium-sized as in the analysis cases, then a shorter list may offer better solutions for the given size of the solution space. The constrained size of the test datasets is also likely the reason that the number of elite ants did not play a significant role in determining the best solution; we would expect them to do so for larger datasets, but this is a topic for future research.

Results for other parametes are more ambiguous. Directions of effects are equivocal for pheromone deposit ($\Delta\tau$), use of ranked ants (Userrank) and pheromone limits (τ_{max} and τ_{min}). For example, where relevant, $\Delta\tau$ displays a decremental effect for Dataset 4 but an incremental effect for Dataset 5, while use of ranked ants has a decremental effect for Datasets 1 and 5, but an incremental effect for Dataset 4. Nonetheless, Shvotba (2005) noted that ranked ants are more useful in terms of speeding up convergence rather than finding the best solution per se. Similarly, τ_{max} and τ_{min} have contrasting effects in terms of between Datasets 1, 2, and 4. Nevertheless, experimental results as shown in Figure 24 have found that, to more evenly distribute node selection probability and avoid stagnation, the lower pheromone trail limits play a more important role than τ_{max}, in line with findings by Stutzle (1999). On the other hand, τ_{max} remains useful for setting the pheromone values during the occasional trail reinitializations whenever the system approaches stagnation or when no improved route has been generated for a certain number of consecutive iterations.

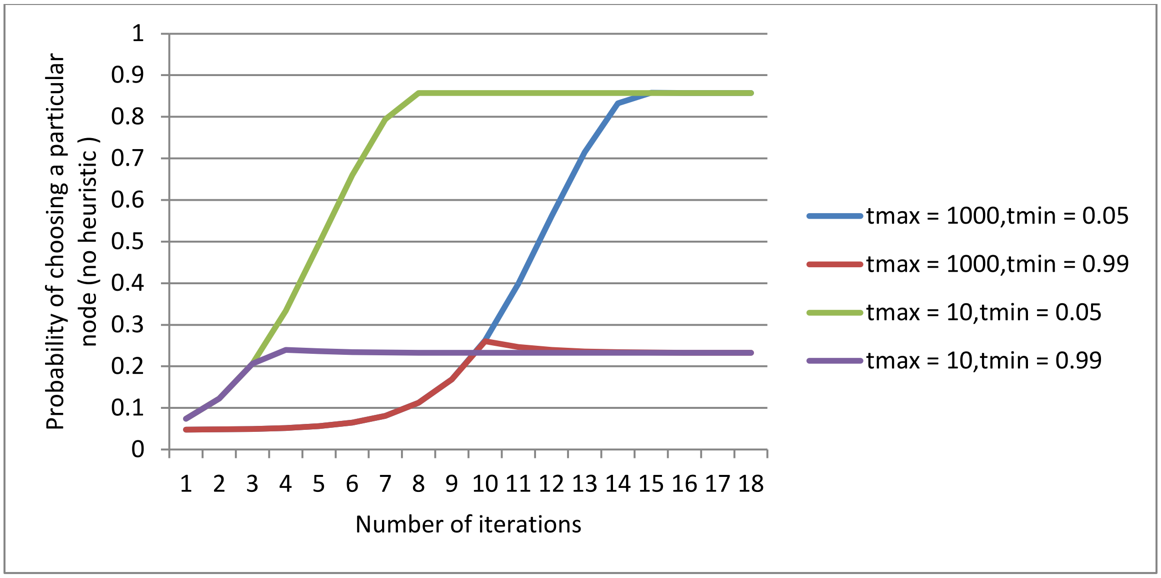
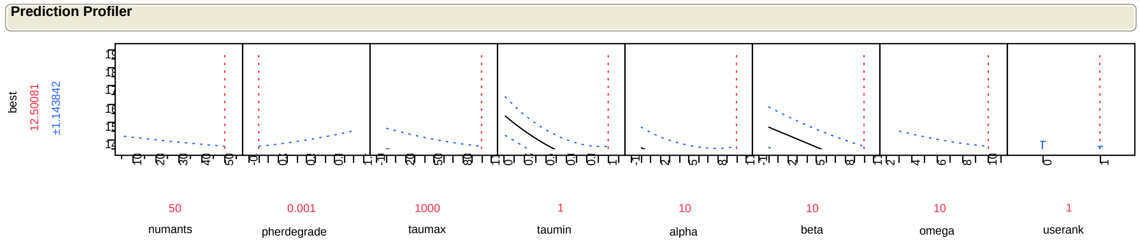


Figure 24. Effect of upper and lower pheromone limits on probability of choosing a node ($\rho = 0.5$).

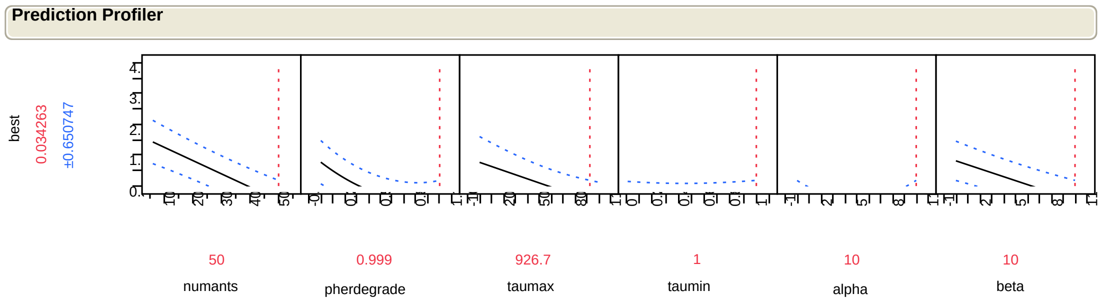
3. Selecting Good Parameter Values

The stepwise-significant parameter values are next tweaked via the JMP prediction profilers to give a sense of the best solution possible and their corresponding parameter values (Figure 25). The ideal settings are generally consistent for most parameters in line with earlier analysis, albeit some parameter settings vary with the datasets. Given the limited range of five test scenarios investigated, future research would be necessary to ascertain the conclusive nature of the relationships between test scenarios and the various parameter settings.

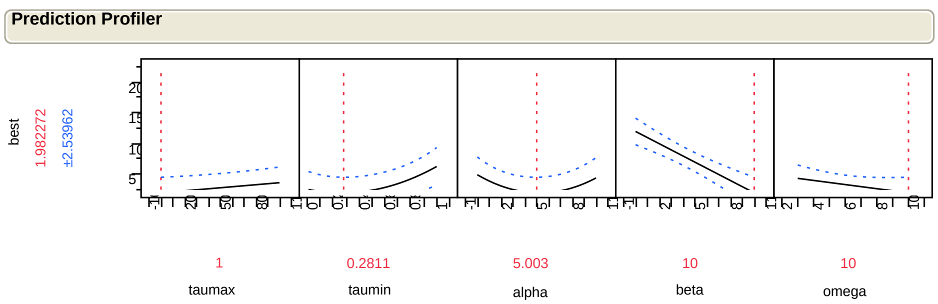
Dataset 1



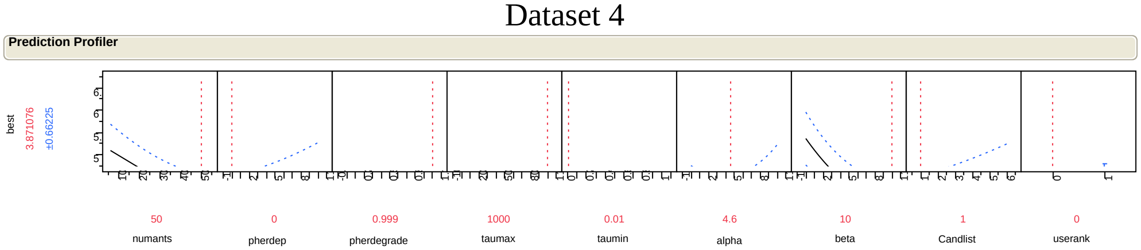
Dataset 2



Dataset 3



Dataset 4



Dataset 5

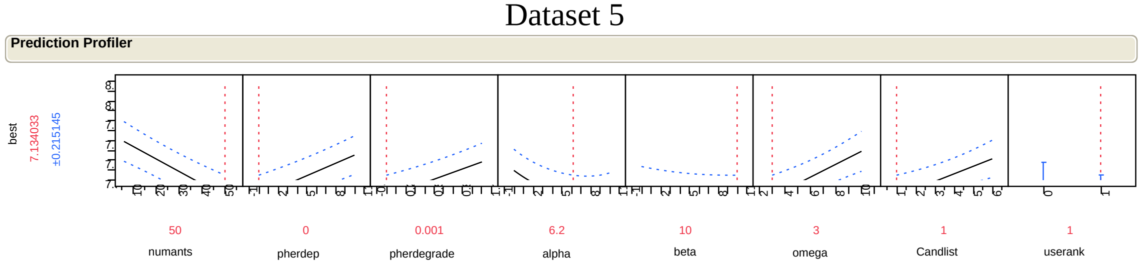


Figure 25. Tweaking of JMP Prediction Profilers to obtain best possible solutions

F. CONCLUSIONS FROM EMPIRICAL ANALYSIS

The empirical analysis has shown that the proposed additive formulation of the algorithm offers generally better solution qualty by adopting wider and longer exploration than the multiplicative formulation.

A good suite of parameter settings to use is one that finds a reasonable balance between too narrow a focus of the search process, which in the worst case, may lead to stagnation behavior, and too weak a guidance of the search, which can cause excessive exploration. Investigation of the relationships between parameters and the number of un-served customers showed that parameter effects may be sensitive to the configuration of the dataset and setting of other parameters.

Analysis of these relationships, as well as taking into consideration findings by Dorigo and Stutzle (2004), Shtovba (2005), Le Louarn, Gendreau, and Potvin (2004), Çatay (2006), and Gajpal & Abad (2009a) who recommends varied values for various ACO implementations based on experience with different types of problems, resulted in the parameter values as suggested in Table 15. They should offer a good balance between maximizing solution quality while keeping convergence time low, for the small and medium-sized datasets studied in these experiments.

Description	**Recommended range**
Number of ants (Numants)	Number of customer nodes
Pheromone weight (α)	0.5 – 2
Heuristic weight (β)	1 – 4
Pheromone degradation (ρ)	0.1
τ_{min}	0.05
τ_{max}	1000
Size of neighborhood heuristic (ant is restricted to Ω closest neighbors)	3 – 8
Size of Candidate List (Candlist)	5 – 20
Pheromones deposit ($\Delta\tau$)	2 – 10
Binary variable whether to use ranked-based AS or not (Userrank)	1
Number of elite ants (σ)	1 – 10

Table 15. Recommended Parameter Value Ranges

VI. CONCLUSION

The objective of this thesis is to develop a novel algorithm solution to solve the complex Overburdened Vehicle Routing Problem (OB-VRP) first formulated in Apte and Heath (2011). The OB-VRP can be construed as an assisted evacuation problem in the context of a short-notice disaster, where the aim is to help government officials provide aid to citizens who cannot self-evacuate, due to lack of private transportation means or disability. The intent is minimize loss of life by developing an evacuation schedule that optimally assigns depot-based vehicles and plans routes to pick up as many "customers" as possible from their homes to a common shelter, within given constraints in terms of the level of disability, vehicle capacities, loading and unloading times, etc. This has been achieved. A summary of the work done and avenues for future work are outlined below.

A. SUMMARY

The thesis first undertook a wide literature review that examined the OB-VRP from both thematic and topical perspectives over the last 60 years. A number of salient conclusions are drawn. Firstly, from a problem definition perspective, the OB-VRP is fundamentally more challenging and complex than past research problems in both humanitarian logistics and VRP fields. From a solution perspective, this has meant that traditional solution approaches such as classical heuristics are not suitable or directly adaptable for the OB-VRP.

Among the many candidate metaheuristic approaches, the thesis has opted for the nature-inspired ACO approach due to its implementation ease and flexibility. Using the Min-Max Ant System (MMAS) as the base algorithm, a number of enhanced features are incorporated to improve solution quality and efficiency, e.g., use of best solution list, elite ants, ranked contribution system, and addition of heuristic procedures during route construction (i.e., nearest-neighborhood and greedy ants).

For empirical analysis, five new stylized datasets are created to mimic a range of test scenarios in terms of size and complexity. The analysis uses an efficient space-filling experimental design method based on Nearly-Orthogonal Latin Hypercubes (NOLH) in testing performance. Two possible algorithmic formulations, Dorigo (2006)'s multiplicative version and a novel additive version, are investigated. The latter is found to offer better solutions by adopting wider and longer exploration than the multiplicative formulation.

The subsequent analysis focused on good metaheuristic parameter settings to use that cater to a broad range of test scenarios. Analysis of parameters effects on solution quality (in terms of number of un-served customers), as well as taking into consideration recommendations from literature, resulted in a set of suggested parameter values that should offer a reasonable balance between maximizing solution quality while keeping convergence time low, for the small and medium-sized datasets studied in the empirical experiments.

B. DIRECTIONS FOR FURTHER RESEARCH

There remain many opportunities for further study into the mechanics of improving the algorithm and tuning the parameter settings to improve its solution quality and speed. Some of these are relatively well-understood and discussed in Chapter V, but others are mentioned below as avenues for future work.

1. Improving Solution Quality

Local Search. The literature on metaheuristics also shows us that a promising approach to improving solution quality is to couple a local search algorithm with a mechanism to generate initial solutions. For example, for the Traveling Salesman Problem (TSP), it is well-known that iterated local search algorithms are currently among the best-performing algorithms. They iteratively apply local search techniques (e.g., 2-opt, 2.5-opt, and 3-opt) to initial solutions that are generated by introducing modification to some locally optimal solutions. In the context of an ACO-based solution approach, once the ants have finished their solution construction, the solutions can be brought to their local optimum by applying a local search routine. Then pheromones are updated on the arcs of the locally optimized solutions. (Dorigo & Stutzle, 2004). Once local search is added, the randomly-generated initial tours may become good enough such that heuristic information is no longer necessary (Dorigo & Stutzle, 2004). Experiments with MMAS and ACS on the TSP have confirmed this conjecture, where very high-quality tours were obtained when used with local search, even without using heuristic information.

Multi-Ant Colony Systems (MACS). An alternative method to optimize routes would be to add a second ant colony system to influence the node selection. In the current implementation, the healthiest solutions are determined by the number of un-served customers they leave; however, this may allow more circuitous or time-consuming routes in the short term, making it difficult for the most efficient node to be chosen. In other words, the "un-served customers" ant colony may unintentionally hem the algorithm into a local solution at the expense of the global optimum. To remedy this, the second ant colony may optimize, for example, individual route times, favoring the shorter routes. The second ant colony would generate its own pheromone matrix and be incorporated into a fusion algorithm (Equations 4.4 and 4.5) with its own weight factor (say, γ). The MACS could easily be used in conjunction with the Local Search routines.

Autotuning of parameters. One of the biggest drawbacks to the ACO-based routines are its sensitivities to the solution size. For example, if the global optimum solution of a particular problem has 100 un-served customers, the amount of pheromones deposited will be sharply diluted. Also affected are the best τ_{max}, τ_{min}, and the optimal pheromone degradation ρ. A promising avenue of further research would be a way to automatically tune the parameters to best match the solution based on size and nature of the problem. An excellent alternative would be finding a formulation for the ACO that removes the sensitivity to the problem size.

2. Improving Solution Speed

The nature of ACO-based routines lends themselves particularly well to parallelization. For example, each ant could run on a separate thread, with pheromone updating occurring real-time; that is, pheromone updating is not critical to run another ant-tour. With the proliferation of high-performance computing available on the Internet, and the low amount of data needed to conduct analysis, it is conceivable that an official in an austere location could set up and run a routine remotely using only a satellite or cellular phone connection.

The algorithm in this thesis was coded in Octave; however, for actual implementation it should be coded in a more user-friendly manner, such as a Microsoft Excel macro or a Java applet, and optimized for maximum speed.

3. Automating Data Gathering

While the code can run large datasets rather efficiently, the bulk of the time spent in the field is likely to be in setting up the dataset. Gathering travel times between nodes quickly explodes into a large affair as the number of nodes grows. An automated program that can compute the travel-time matrix from points on a map would serve well to make the program as a whole more usable.

4. Morality of Solutions

While the code optimizes the number of people that are saved, it does not differentiate in any fashion among the people. The societal implications of some of the solutions should be taken into account. For example, since a stretcher-bound patient takes longer to load and occupies more room than a person who is only slightly hobbled, the routine will favor picking up those least-disabled. The best analogy for the use of the routine is a triage center, where the worst-injured are ignored in order to maximize the number who are saved; this may prove unpopular in aggregate. Imagine the furor if the government publicly acknowledged ignoring wheelchair-bound people for those who were slightly ambulatory. A future evolution of the algorithm may want to weight certain disabilities over others to satisfy public opinion.

C. FINAL REMARKS

The field of Humanitarian Assistance and Disaster Relief (HADR) continues to gain interest and importance. By offering a way of solving an evacuation problem, the algorithmic techniques developed in this body of work can be integrated into a larger optimization tools framework and would serve as a discourse toward improving the nature of HADR operations as a whole.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. DETAILED DATASET SPECIFICATIONS

Dataset 1

No.	Type	X-coord	Y-coord	Customer disability level	Customer loading time
1	Shelter	0	0		
2	Customer	2	2	1	75
3	Customer	-2	2	1	60
4	Customer	-2	-2	1	10
5	Customer	2	-2	1	15
6	Customer	4	0	1	18
7	Customer	3	1	1	5
8	Customer	5	4	1	2
9	Customer	1	-2	1	1
10	Customer	12	-1	1	17
11	Customer	6	12	1	40
12	Customer	8	0	1	34
13	Customer	3	2	1	2
14	Customer	-6	-1	1	5
15	Customer	-5	-4	1	8
16	Customer	4	-6	1	18
17	Customer	-5	7	1	62
18	Customer	-6	4	1	63
19	Customer	2	3	1	5
20	Customer	-1	-2	1	6
21	Customer	1	-7	1	2
22	Customer	2	7	1	18
23	Customer	10	-3	1	90
24	Customer	12	2	1	33
25	Customer	-10	-1	1	1
26	Customer	-4	1	1	2
27	Customer	-2	5	1	18
28	Customer	3	-6	1	17
29	Customer	8	3	1	15
30	Customer	0	0	1	20
31	Customer	-5	2	1	2
32	Customer	-8	3	1	23
33	Customer	9	0	1	6

34	Customer	4	0	1	7
35	Customer	8	7	1	12
36	Customer	2	5	1	90
37	Customer	3	3	1	56
38	Customer	17	4	1	2
39	Customer	2	7	1	78
40	Customer	9	2	1	18
41	Customer	-5	-6	1	4
42	Depot	-7	2		
43	Depot	-2	-12		

Vehicle	**Capacity for disability level 1**
1	6
2	8

Maximum number of tours per vehicle, *R*	**Total time available, *T***
2	200

Dataset 2

No.	**Type**	**X-coord**	**Y-coord**	**Customer disability level**	**Customer loading time**
1	Shelter	**27**	14		
2	Customer	60	36	2	23
3	Customer	25	42	1	46
4	Customer	22	51	1	49
5	Customer	37	50	3	46
6	Customer	16	57	1	3
7	Customer	40	24	3	36
8	Customer	28	18	1	9
9	Customer	8	40	3	34
10	Customer	44	49	1	10
11	Customer	38	19	3	7
12	Customer	50	28	3	21
13	Customer	29	1	1	20
14	Customer	30	53	1	24
15	Customer	50	2	1	36
16	Customer	53	59	3	1
17	Customer	9	53	2	17
18	Customer	35	50	2	47
19	Customer	23	5	2	43

20	Customer	28	56	3	39
21	Customer	43	16	3	25
22	Customer	45	38	3	38
23	Customer	49	34	2	15
24	Customer	47	48	3	41
25	Customer	43	44	3	13
26	Customer	7	57	2	45
27	Customer	31	33	2	21
28	Customer	15	27	3	0
29	Customer	44	43	1	32
30	Customer	25	40	2	34
31	Customer	22	52	2	40
32	Customer	43	21	2	43
33	Customer	34	20	2	44
34	Customer	52	36	3	19
35	Customer	32	23	2	48
36	Customer	9	30	2	36
37	Customer	28	46	3	15
38	Customer	42	24	1	35
39	Customer	8	9	1	24
40	Customer	46	3	3	9
41	Customer	2	0	2	22
42	Depot	18	31		
43	depot	19	8		
44	Depot	44	30		
45	Depot	11	33		

Vehicle	**Capacity for disability level 1**	**Capacity for disability level 2**	**Capacity for disability level 3**
1	3	6	0
2	0	5	4
3	1	5	1
4	6	1	5

Maximum number of tours per vehicle, *R*	**Total time available, *T***
5	500

Dataset 3

No.	Type	X-coord	Y-coord	Customer disability level	Customer loading time
1	Shelter	19	8		
2	Customer	14	17	1	6
3	Customer	9	6	2	12
4	Customer	11	11	2	12
5	Customer	11	18	2	12
6	Customer	13	5	1	6
7	Customer	9	10	2	12
8	Customer	19	7	1	6
9	Customer	11	18	1	6
10	Customer	6	12	2	12
11	Customer	0	6	2	12
12	Customer	0	15	2	12
13	Customer	4	5	1	6
14	Customer	18	16	1	6
15	Customer	20	15	1	6
16	Customer	6	14	2	12
17	Customer	15	16	2	12
18	Customer	18	7	1	6
19	Customer	17	11	2	12
20	Customer	5	11	1	6
21	Customer	11	3	1	6
22	Customer	19	15	2	12
23	Customer	14	17	2	12
24	Customer	10	6	1	6
25	Customer	16	16	2	12
26	Customer	9	18	1	6
27	Customer	3	5	1	6
28	Customer	13	2	2	12
29	Customer	9	19	2	12
30	Customer	2	12	2	12
31	Customer	12	18	2	12
32	Customer	3	11	2	12
33	Customer	18	18	1	6
34	Customer	4	1	2	12
35	Customer	19	6	2	12
36	Customer	17	6	1	6

37	Customer	13	14	2	12
38	Customer	2	6	2	12
39	Customer	18	6	2	12
40	Customer	8	2	2	12
41	Customer	17	20	2	12
42	Customer	8	10	1	6
43	Customer	7	16	1	6
44	Customer	2	5	2	12
45	Customer	13	18	1	6
46	Customer	12	17	1	6
47	Customer	7	1	1	6
48	Customer	16	14	1	6
49	Customer	13	11	2	12
50	Customer	19	20	1	6
51	Customer	13	7	2	12
52	Customer	0	4	2	12
53	Customer	19	16	1	6
54	Customer	18	20	1	6
55	Customer	14	18	1	6
56	Customer	9	14	2	12
57	Customer	13	14	1	6
58	Customer	3	0	2	12
59	Customer	16	13	2	12
60	Customer	7	17	1	6
61	Customer	5	5	1	6
62	Customer	17	9	1	6
63	Customer	15	16	2	12
64	Customer	9	14	1	6
65	Customer	14	12	1	6
66	Customer	1	15	2	12
67	Customer	4	13	2	12
68	Customer	18	3	2	12
69	Customer	9	18	2	12
70	Customer	2	10	1	6
71	Customer	18	10	1	6
72	Customer	3	8	2	12
73	Customer	2	20	1	6
74	Customer	0	11	2	12
75	Customer	0	20	1	6
76	Customer	14	8	2	12

77	Customer	16	6	2	12
78	Customer	3	18	2	12
79	Customer	1	1	2	12
80	Customer	14	17	1	6
81	Customer	6	14	1	6
82	Customer	17	12	2	12
83	Customer	16	0	2	12
84	Customer	5	9	1	6
85	Customer	4	10	1	6
86	Customer	16	13	2	12
87	Customer	13	6	2	12
88	Customer	6	19	1	6
89	Customer	14	8	2	12
90	Customer	13	14	2	12
91	Customer	11	17	2	12
92	Customer	14	17	1	6
93	Customer	14	3	1	6
94	Customer	17	13	2	12
95	Customer	12	5	2	12
96	Customer	18	19	1	6
97	Customer	12	12	2	12
98	Customer	8	19	1	6
99	Customer	5	10	1	6
100	Customer	15	2	2	12
101	Customer	3	7	2	12
102	Depot	18	5		
103	Depot	12	1		
104	Depot	17	12		
105	Depot	8	7		

Vehicle	**Capacity for disability level 1**	**Capacity for disability level 2**
1	1	5
2	3	4
3	3	4
4	1	3

Maximum number of tours per vehicle, *R*	**Total time available, *T***
6	400

Dataset 4

No.	Type	X-coord	Y-coord	Customer disability level	Customer loading time
1	Shelter	0	0		
2	Customer	1	1	1	1
3	Customer	2	2	1	1
4	Customer	-2	-2	1	1
5	Customer	-1	-1	1	1
6	Customer	3	3	1	1
7	Customer	-3	-3	1	1
8	Customer	4	4	1	1
9	Customer	-4	-4	1	1
10	Customer	-5	-5	1	1
11	Customer	5	5	1	1
12	Customer	8	-8	1	1
13	Customer	10	-10	1	1
14	Customer	-10	10	1	1
15	Customer	-8	8	1	1
16	Customer	15	0	1	1
17	Depot	-6	-6		
18	Depot	6	6		

Vehicle	Capacity for disability level 1
1	5
2	5

Maximum number of tours per vehicle, *R*	Total time available, *T*
4	15

Dataset 5

No.	Type	X-coord	Y-coord	Customer disability level	Customer loading time
1	Shelter	27	4		
2	Customer	22	11	1	10
3	Customer	10	7	2	20
4	Customer	26	7	1	10
5	Customer	19	5	2	20
6	Customer	4	10	1	10
7	Customer	5	22	1	10

8	Customer	27	7	1	10
9	Customer	24	0	1	10
10	Customer	3	11	2	20
11	Customer	9	21	1	10
12	Customer	11	21	2	20
13	Customer	29	2	2	20
14	Customer	11	2	1	10
15	Customer	22	7	1	10
16	Customer	9	24	2	20
17	Customer	16	18	2	20
18	Customer	18	6	2	20
19	Customer	26	5	1	10
20	Customer	15	9	1	10
21	Customer	23	21	2	20
22	Depot	16	1		
23	Depot	27	14		
24	Depot	17	15		

Vehicle	**Capacity for disability level 1**	**Capacity for disability level 2**
1	2	2
2	1	1
3	1	3

Maximum number of tours per vehicle, *R*	**Total time available, *T***
6	300

APPENDIX B. NOLH DESIGN SPECIFICATIONS

Parameter	iter	m	$\Delta\tau$	ρ	τ_{max}	τ_{min}	α	β	Ω	σ	w	Best
Lowest Value	5	6	0.01	0.001	1	0.01	0	0	3	0	1	0
Highest Value	100	50	10	0.999	1000	1	10	10	10	10	6	1
Decimal places per interval	0	0	1	3	0	2	1	1	0	0	0	0
Scenario 1	73	8	3.6	0.328	126	0.77	8.0	4.8	10	7	4	1
Scenario 2	96	38	1.1	0.422	344	0.26	5.5	7.5	8	9	5	0
Scenario 3	90	22	9.5	0.219	298	0.86	1.6	4.5	6	6	5	1
Scenario 4	66	45	7.2	0.453	63	0.43	2.7	2.7	4	10	6	1
Scenario 5	93	27	1.9	0.017	95	0.16	2.5	6.1	7	5	1	1
Scenario 6	55	47	2.4	0.484	157	0.66	1.4	8.6	9	1	3	1
Scenario 7	78	14	5.3	0.032	251	0.33	8.9	4.2	4	4	3	1
Scenario 8	82	40	9.2	0.313	376	0.92	7.0	0.2	5	3	1	1
Scenario 9	70	7	0.2	0.812	407	0.61	3.9	0.9	9	3	5	0
Scenario 10	97	36	4.8	0.765	1	0.29	9.2	3.8	8	0	4	0
Scenario 11	54	7	9.7	0.531	204	0.44	3.1	6.9	3	2	5	0
Scenario 12	99	29	6.9	0.921	173	0.07	3.8	8.1	5	4	4	0
Scenario 13	57	16	3.3	0.640	391	0.49	1.9	2.8	10	10	2	0
Scenario 14	79	30	4.2	0.890	266	0.97	0.5	0.0	7	5	2	0
Scenario 15	60	20	7.8	0.952	438	0.23	4.7	9.4	4	8	1	0
Scenario 16	69	31	5.5	0.718	141	0.95	10.0	5.6	6	8	3	0
Scenario 17	87	25	3.9	0.126	672	1.00	7.8	9.7	7	6	3	0
Scenario 18	58	44	2.7	0.344	579	0.46	8.3	8.0	6	7	2	0
Scenario 19	75	23	5.6	0.297	781	0.89	0.3	4.1	7	8	3	0
Scenario 20	63	38	8.6	0.063	547	0.18	4.4	1.7	9	9	1	0
Scenario 21	84	15	4.1	0.095	984	0.38	0.2	7.0	5	1	3	0
Scenario 22	81	42	0.0	0.157	516	0.71	4.8	8.9	4	3	6	0
Scenario 23	100	24	8.0	0.251	969	0.13	7.2	1.3	7	3	4	0
Scenario 24	61	50	8.3	0.375	688	0.54	5.8	2.2	10	1	5	0
Scenario 25	64	17	0.9	0.594	813	0.74	9.4	1.6	4	2	2	1
Scenario 26	76	34	1.6	0.999	766	0.20	6.4	3.6	6	5	1	1
Scenario 27	85	10	7.5	0.796	532	0.91	3.3	9.5	8	3	2	1
Scenario 28	91	43	6.3	0.827	922	0.32	1.3	6.6	8	4	3	1
Scenario 29	94	19	1.3	0.609	641	0.21	4.1	0.8	5	9	5	1
Scenario 30	72	47	3.4	0.734	891	0.98	3.4	5.3	3	8	4	1
Scenario 31	67	12	7.0	0.562	953	0.41	7.7	7.7	9	10	5	1
Scenario 32	88	35	9.4	0.859	719	0.64	9.1	6.7	8	6	5	1
Scenario 33	53	28	5.0	0.500	501	0.51	5.0	5.0	7	5	4	1
Scenario 34	32	48	6.4	0.672	875	0.24	2.0	5.2	3	3	3	0
Scenario 35	9	18	8.9	0.578	657	0.75	4.5	2.5	5	1	2	1
Scenario 36	15	34	0.5	0.781	703	0.15	8.4	5.5	7	4	2	0
Scenario 37	39	11	2.8	0.547	938	0.58	7.3	7.3	9	0	1	0

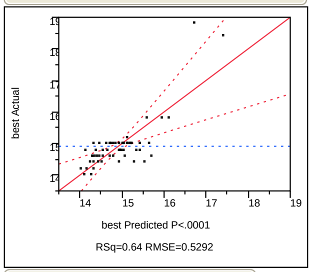
Scenario 38	12	29	8.1	0.983	906	0.85	7.5	3.9	6	5	6	0
Scenario 39	50	9	7.7	0.516	844	0.35	8.6	1.4	4	9	4	0
Scenario 40	27	42	4.7	0.968	750	0.68	1.1	5.8	9	6	4	0
Scenario 41	23	16	0.8	0.687	625	0.09	3.0	9.8	8	7	6	0
Scenario 42	35	49	9.8	0.188	594	0.40	6.1	9.1	4	7	2	1
Scenario 43	8	20	5.2	0.235	1000	0.72	0.8	6.3	5	10	3	1
Scenario 44	51	49	0.3	0.469	797	0.57	6.9	3.1	10	8	2	1
Scenario 45	6	27	3.1	0.079	828	0.94	6.3	1.9	8	6	3	1
Scenario 46	48	40	6.7	0.360	610	0.52	8.1	7.2	3	0	5	1
Scenario 47	26	26	5.8	0.110	735	0.04	9.5	10.0	6	5	5	1
Scenario 48	45	36	2.2	0.048	563	0.78	5.3	0.6	9	2	6	1
Scenario 49	36	25	4.5	0.282	860	0.06	0.0	4.4	7	2	4	1
Scenario 50	18	31	6.1	0.874	329	0.01	2.2	0.3	6	4	4	1
Scenario 51	47	12	7.3	0.656	422	0.55	1.7	2.0	7	3	5	1
Scenario 52	30	33	4.4	0.703	220	0.12	9.7	5.9	6	2	4	1
Scenario 53	42	18	1.4	0.937	454	0.83	5.6	8.3	4	1	6	1
Scenario 54	21	41	5.9	0.905	17	0.63	9.8	3.0	8	9	4	1
Scenario 55	24	14	10.0	0.843	485	0.30	5.2	1.1	9	7	1	1
Scenario 56	5	32	2.0	0.750	32	0.88	2.8	8.8	6	7	3	1
Scenario 57	44	6	1.7	0.625	313	0.47	4.2	7.8	3	9	2	1
Scenario 58	41	39	9.1	0.406	188	0.27	0.6	8.4	9	8	5	0
Scenario 59	29	23	8.4	0.001	235	0.81	3.6	6.4	7	5	6	0
Scenario 60	20	46	2.5	0.204	469	0.10	6.7	0.5	5	8	5	0
Scenario 61	14	13	3.8	0.173	79	0.69	8.8	3.4	5	6	4	0
Scenario 62	11	37	8.8	0.391	360	0.80	5.9	9.2	8	1	2	0
Scenario 63	33	9	6.6	0.266	110	0.03	6.6	4.7	10	2	3	0
Scenario 64	38	45	3.0	0.438	48	0.60	2.3	2.3	4	0	2	0
Scenario 65	17	21	0.6	0.141	282	0.37	0.9	3.3	5	4	2	0

APPENDIX C. FULL JMP REGRESSION OUTPUTS

Dataset 1

Response best

Actual by Predicted Plot



Summary of Fit

RSquare	0.637825
RSquare Adj	0.5171
Root Mean Square Error	0.52921
Mean of Response	14.91385
Observations (or Sum Wgts)	65

Analysis of Variance

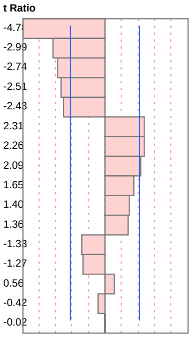
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	16	23.674481	1.47966	5.2833
Error	48	13.443058	0.28006	Prob > F
C. Total	64	37.117538		<.0001

Parameter Estimates

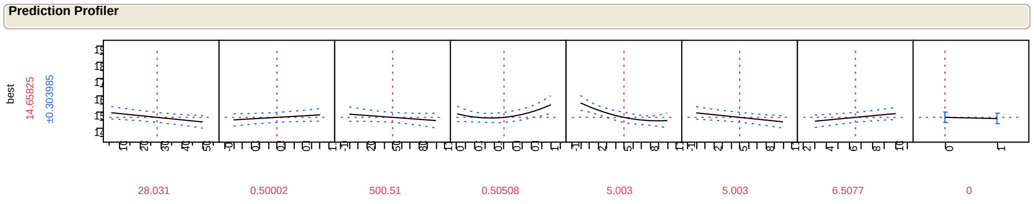
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	15.252414	0.386206	39.49	<.0001
numants	-0.012788	0.005087	-2.51	0.0153
pherdegrade	0.306539	0.224726	1.36	0.1789
taumax	-9.444e-5	0.000224	-0.42	0.6758
taumin	-0.003607	0.227471	-0.02	0.9874
alpha	-0.107355	0.022444	-4.78	<.0001
beta	-0.054674	0.022525	-2.43	0.0190
omega	0.066647	0.031867	2.09	0.0418
userank[0]	0.0370168	0.066671	0.56	0.5813
(pherdegrade-0.50002)*(taumin-0.50508)	1.1947539	0.724964	1.65	0.1059
(taumax-500.508)*(taumin-0.50508)	-0.002433	0.000815	-2.99	0.0044
(taumax-500.508)*userank[0]	-0.000296	0.000233	-1.27	0.2093
(taumin-0.50508)*(beta-5.00308)	-0.098633	0.073963	-1.33	0.1886
(taumin-0.50508)*userank[0]	0.5624976	0.243743	2.31	0.0254
(beta-5.00308)*(omega-6.50769)	-0.033823	0.012356	-2.74	0.0087
(taumin-0.50508)*(taumin-0.50508)	2.0098023	0.888236	2.26	0.0282
(alpha-5.00308)*(alpha-5.00308)	0.0138277	0.009889	1.40	0.1685

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
alpha	-0.107355	0.022444	-4.7	<.0001
(taumax-500.508)*(taumin-0.50508)	-0.002433	0.000815	-2.9	0.0044
(beta-5.00308)*(omega-6.50769)	-0.033823	0.012356	-2.74	0.0087
numants	-0.012788	0.005087	-2.51	0.0153
beta	-0.054674	0.022525	-2.43	0.0190
(taumin-0.50508)*userank[0]	0.5624976	0.243743	2.31	0.0254
(taumin-0.50508)*(taumin-0.50508)	2.0098023	0.888236	2.26	0.0282
omega	0.066647	0.031867	2.09	0.0418
(pherdegrade-0.50002)*(taumin-0.50508)	1.1947539	0.724964	1.65	0.1059
(alpha-5.00308)*(alpha-5.00308)	0.0138277	0.009889	1.40	0.1685
pherdegrade	0.306539	0.224726	1.36	0.1789
(taumin-0.50508)*(beta-5.00308)	-0.098633	0.073963	-1.33	0.1886
(taumax-500.508)*userank[0]	-0.000296	0.000233	-1.27	0.2093
userank[0]	0.0370168	0.066671	0.56	0.5813
taumax	-9.444e-5	0.000224	-0.42	0.6758
taumin	-0.003607	0.227471	-0.02	0.9874



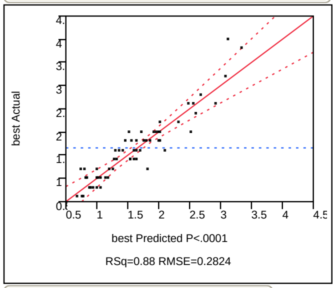
Prediction Profiler



Dataset 2

Response best

Actual by Predicted Plot



Summary of Fit

RSquare	0.876519
RSquare Adj	0.841945
Root Mean Square Error	0.282424
Mean of Response	1.658462
Observations (or Sum Wgts)	65

Analysis of Variance

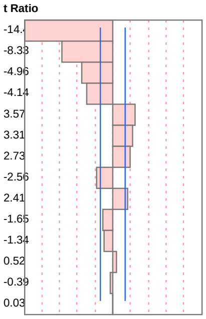
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	14	28.309681	2.02212	25.3515
Error	50	3.988166	0.07976	Prob > F
C. Total	64	32.297846		<.0001

Parameter Estimates

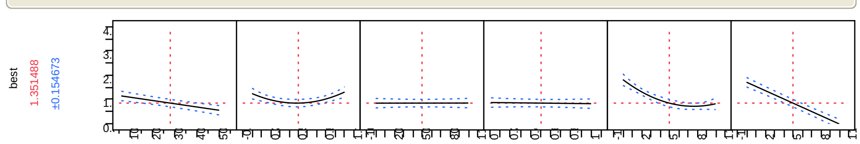
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	3.0814267	0.172888	17.82	<.0001
numants	-0.013456	0.002714	-4.96	<.0001
pherdegrade	0.0621312	0.11975	0.52	0.6062
taumax	3.9189e-6	0.00012	0.03	0.9740
taumin	-0.047386	0.120753	-0.39	0.6964
alpha	-0.099588	0.011958	-8.33	<.0001
beta	-0.172617	0.011958	-14.44	<.0001
(numants-28.0308)*(pherdegrade-0.50002)	-0.05945	0.014347	-4.14	0.0001
(taumax-500.508)*(taumin-0.50508)	-0.001106	0.000431	-2.56	0.0134
(taumax-500.508)*(alpha-5.00308)	-7.118e-5	0.000043	-1.65	0.1043
(taumax-500.508)*(beta-5.00308)	-8.724e-5	6.529e-5	-1.34	0.1875
(taumin-0.50508)*(alpha-5.00308)	0.1005417	0.041658	2.41	0.0195
(alpha-5.00308)*(beta-5.00308)	0.0162713	0.005952	2.73	0.0086
(pherdegrade-0.50002)*(pherdegrade-0.50002)	1.7107513	0.478801	3.57	0.0008
(alpha-5.00308)*(alpha-5.00308)	0.0188542	0.005693	3.31	0.0017

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t
beta	-0.172617	0.011958	-14.4	<.0001
alpha	-0.099588	0.011958	-8.33	<.0001
numants	-0.013456	0.002714	-4.96	<.0001
(numants-28.0308)*(pherdegrade-0.50002)	-0.05945	0.014347	-4.14	0.0001
(pherdegrade-0.50002)*(pherdegrade-0.50002)	1.7107513	0.478801	3.57	0.0008
(alpha-5.00308)*(alpha-5.00308)	0.0188542	0.005693	3.31	0.0017
(alpha-5.00308)*(beta-5.00308)	0.0162713	0.005952	2.73	0.0086
(taumax-500.508)*(taumin-0.50508)	-0.001106	0.000431	-2.56	0.0134
(taumin-0.50508)*(alpha-5.00308)	0.1005417	0.041658	2.41	0.0195
(taumax-500.508)*(alpha-5.00308)	-7.118e-5	0.000043	-1.65	0.1043
(taumax-500.508)*(beta-5.00308)	-8.724e-5	6.529e-5	-1.34	0.1875
pherdegrade	0.0621312	0.11975	0.52	0.6062
taumin	-0.047386	0.120753	-0.39	0.6964
taumax	3.9189e-6	0.00012	0.03	0.9740



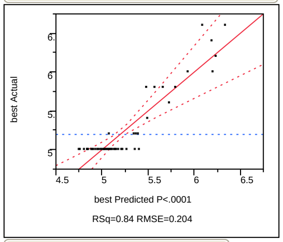
Prediction Profiler



Dataset 4

Response best

Actual by Predicted Plot



Summary of Fit

RSquare	0.844472
RSquare Adj	0.763005
Root Mean Square Error	0.20401
Mean of Response	5.196923
Observations (or Sum Wgts)	65

Analysis of Variance

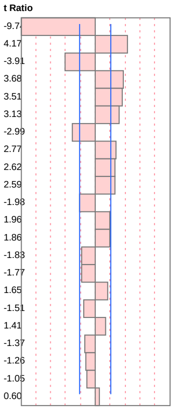
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	22	9.491346	0.431425	10.3658
Error	42	1.748038	0.041620	Prob > F
C. Total	64	11.239385		<.0001

Parameter Estimates

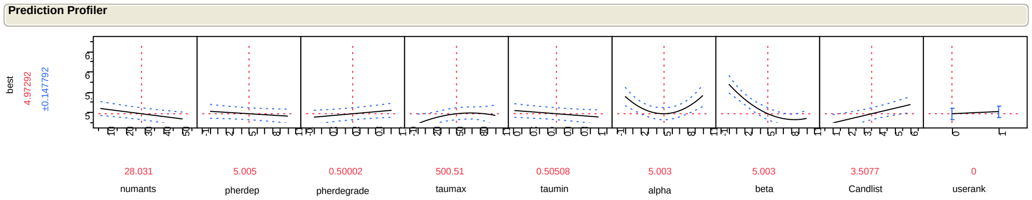
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	5.6768728	0.148438	38.24	<.0001
numants	-0.005873	0.001961	-2.99	0.0046
pherdep	-0.011821	0.008646	-1.37	0.1788
pherdegrade	0.1698374	0.086649	1.96	0.0566
taumax	5.1783e-5	8.658e-5	0.60	0.5530
taumin	-0.160588	0.087693	-1.83	0.0742
alpha	-0.03386	0.008654	-3.91	0.0003
beta	-0.084351	0.008663	-9.74	<.0001
Candlist	0.0309212	0.016594	1.86	0.0694
userank[0]	-0.026954	0.025607	-1.05	0.2986
(numants-28.0308)*(pherdegrade-0.50002)	-0.01936	0.01284	-1.51	0.1391
(numants-28.0308)*(taumin-0.50508)	0.0249629	0.009642	2.59	0.0132
(pherdep-5.00462)*(beta-5.00308)	0.0117169	0.003741	3.13	0.0032
(pherdep-5.00462)*(Candlist-3.50769)	-0.012424	0.006285	-1.98	0.0547
(taumax-500.508)*(taumin-0.50508)	-0.000577	0.000326	-1.77	0.0838
(taumax-500.508)*userank[0]	0.0001536	0.000109	1.41	0.1652
(taumin-0.50508)*(alpha-5.00308)	0.0502044	0.030352	1.65	0.1056
(alpha-5.00308)*(beta-5.00308)	0.0162017	0.00461	3.51	0.0011
(alpha-5.00308)*userank[0]	0.036172	0.009836	3.68	0.0007
(Candlist-3.50769)*userank[0]	0.060065	0.021652	2.77	0.0082
(taumax-500.508)*(taumax-500.508)	-5.77e-7	4.584e-7	-1.26	0.2150
(alpha-5.00308)*(alpha-5.00308)	0.0175901	0.004219	4.17	0.0001
(beta-5.00308)*(beta-5.00308)	0.0122354	0.004676	2.62	0.0123

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
beta	-0.084351	0.008663	-9.7	<.0001
(alpha-5.00308)*(alpha-5.00308)	0.0175901	0.004219	4.17	0.0001
alpha	-0.03386	0.008654	-3.91	0.0003
(alpha-5.00308)*userank[0]	0.036172	0.009836	3.68	0.0007
(alpha-5.00308)*(beta-5.00308)	0.0162017	0.00461	3.51	0.0011
(pherdep-5.00462)*(beta-5.00308)	0.0117169	0.003741	3.13	0.0032
numants	-0.005873	0.001961	-2.99	0.0046
(Candlist-3.50769)*userank[0]	0.060065	0.021652	2.77	0.0082
(beta-5.00308)*(beta-5.00308)	0.0122354	0.004676	2.62	0.0123
(numants-28.0308)*(taumin-0.50508)	0.0249629	0.009642	2.59	0.0132
(pherdep-5.00462)*(Candlist-3.50769)	-0.012424	0.006285	-1.98	0.0547
pherdegrade	0.1698374	0.086649	1.96	0.0566
Candlist	0.0309212	0.016594	1.86	0.0694
taumin	-0.160588	0.087693	-1.83	0.0742
(taumax-500.508)*(taumin-0.50508)	-0.000577	0.000326	-1.77	0.0838
(taumin-0.50508)*(alpha-5.00308)	0.0502044	0.030352	1.65	0.1056
(numants-28.0308)*(pherdegrade-0.50002)	-0.01936	0.01284	-1.51	0.1391
(taumax-500.508)*userank[0]	0.0001536	0.000109	1.41	0.1652
pherdep	-0.011821	0.008646	-1.37	0.1788
(taumax-500.508)*(taumax-500.508)	-5.77e-7	4.584e-7	-1.26	0.2150
userank[0]	-0.026954	0.025607	-1.05	0.2986
taumax	5.1783e-5	8.658e-5	0.60	0.5530



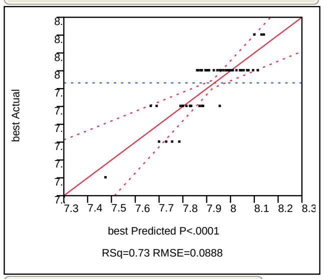
Prediction Profiler



Dataset 5

Response best

Actual by Predicted Plot



Summary of Fit

RSquare	0.732
RSquare Adj	0.635064
Root Mean Square Error	0.088776
Mean of Response	7.932308
Observations (or Sum Wgts)	65

Analysis of Variance

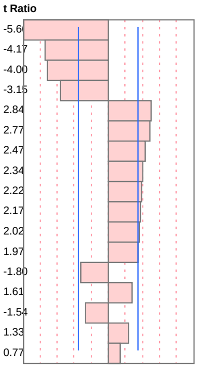
Source	DF	Sum of Squares	Mean Square	F Ratio
Model	17	1.0117366	0.059514	7.5514
Error	47	0.3704173	0.007881	Prob > F
C. Total	64	1.3821538		<.0001

Parameter Estimates

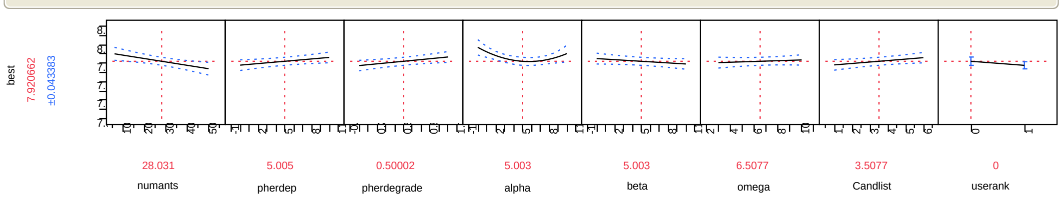
Term	Estimate	Std Error	t Ratio	Prob>|t|
Intercept	7.9717517	0.065228	122.21	<.0001
numants	-0.004835	0.000854	-5.66	<.0001
pherdep	0.0083437	0.003762	2.22	0.0314
pherdegrade	0.0931283	0.037714	2.47	0.0172
alpha	-0.015694	0.003766	-4.17	0.0001
beta	-0.005824	0.003784	-1.54	0.1305
omega	0.0041357	0.005348	0.77	0.4432
Candlist	0.0156549	0.00722	2.17	0.0352
userank[0]	0.021938	0.01111	1.97	0.0542
(numants-28.0308)*(pherdep-5.00462)	0.000463	0.000288	1.61	0.1144
(numants-28.0308)*(Candlist-3.50769)	0.0017243	0.000608	2.84	0.0067
(numants-28.0308)*userank[0]	0.0011725	0.00088	1.33	0.1891
(pherdep-5.00462)*(Candlist-3.50769)	-0.009284	0.002319	-4.00	0.0002
(pherdegrade-0.50002)*(omega-6.50769)	-0.073264	0.023269	-3.15	0.0028
(alpha-5.00308)*userank[0]	0.0089431	0.004424	2.02	0.0489
(beta-5.00308)*(omega-6.50769)	0.0049153	0.002103	2.34	0.0238
(beta-5.00308)*(Candlist-3.50769)	-0.004782	0.002656	-1.80	0.0783
(alpha-5.00308)*(alpha-5.00308)	0.0046814	0.001692	2.77	0.0081

Sorted Parameter Estimates

Term	Estimate	Std Error	t Ratio	Prob>|t|
numants	-0.004835	0.000854	-5.66	<.0001
alpha	-0.015694	0.003766	-4.17	0.0001
(pherdep-5.00462)*(Candlist-3.50769)	-0.009284	0.002319	-4.00	0.0002
(pherdegrade-0.50002)*(omega-6.50769)	-0.073264	0.023269	-3.15	0.0028
(numants-28.0308)*(Candlist-3.50769)	0.0017243	0.000608	2.84	0.0067
(alpha-5.00308)*(alpha-5.00308)	0.0046814	0.001692	2.77	0.0081
pherdegrade	0.0931283	0.037714	2.47	0.0172
(beta-5.00308)*(omega-6.50769)	0.0049153	0.002103	2.34	0.0238
pherdep	0.0083437	0.003762	2.22	0.0314
Candlist	0.0156549	0.00722	2.17	0.0352
(alpha-5.00308)*userank[0]	0.0089431	0.004424	2.02	0.0489
userank[0]	0.021938	0.01111	1.97	0.0542
(beta-5.00308)*(Candlist-3.50769)	-0.004782	0.002656	-1.80	0.0783
(numants-28.0308)*(pherdep-5.00462)	0.000463	0.000288	1.61	0.1144
beta	-0.005824	0.003784	-1.54	0.1305
(numants-28.0308)*userank[0]	0.0011725	0.00088	1.33	0.1891
omega	0.0041357	0.005348	0.77	0.4432



Prediction Profiler



LIST OF REFERENCES

Aarts, E. H. L., & Ten Eikelder, H. M. M. (2002). Simulated annealing. In P. M. Pardalos & M. G. C. Resende (Eds.), *Handbook of Applied Optimization* (pp. 209–220). New York: Oxford University Press.

Aarts, E. H. L., & Lenstra, J. K. (1996). *Local Search in Combinatorial Optimization*. Chichester, UK: John Wiley.

Agarwal, Y., Mathur, K., & Salkin, H. M. (1989). A set-partitioning-based exact algorithm for the vehicle routing problem. *Networks, 19*, 731–749.

Ahmad, S., & Simonovic, S. P. (2006). An intelligent decision support system for management of floods. *Water Resources Management Journal, 20*(3), 391–410.

Ai, T. J., & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers and Operations Research, 36*(5), 1693–1702.

Anbuudayasankar, S. P., & Mohandas, K. (2008). Mixed-integer linear programming for vehicle routing problem with simultaneous delivery and pick-up with maximum route-length. *International Journal of Applied Management and Technology, 1*(6), 31–52.

Angelelli, E., & Mansini, R. (2004). The vehicle routing problem with time windows and simultaneous pick-up and delivery. In A. Klose , M. G. Speranza, & L. N. Van Wassenhove (Eds.), *Quantitative Approaches to Distribution Logistics and Supply Chain Management* (pp. 249–267). Berlin: Springer-Verlag.

Apte, A. (2009). Humanitarian Logistics: A new field of research and action. *Foundations and Trends in Technology, Information and Operations Management, 3*(1), 1–100.

Apte, A., & Heath, S. (2011). Request and response processes for Department of Defense support during domestic disasters. *Journal of Homeland Security and Emergency Management, 8*(1), Article 17.

Baker, B. M., & Ayechew, M. A. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research. 30*, 787-800.

Bakuli, D. L., & Smith, J. M. (1996). Resource allocation in state-dependent emergency evacuation networks. *European Journal of Operation Research, 89,* 543–555.

Baldacci, R., Hadjiconstantinou, E., & Mingozzi, A. (2003). An exact algorithm for the traveling salesman problem with deliveries and collections, *Networks, 42*, 26–41.

Baldacci, R., Battarra, M., & Vigo, D. (2008). Routing a heterogeneous fleet of vehicles. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 3–27). Berlin: Springer.

Balinski, M., & Quandt, R. (1964). On an integer program for a delivery problem. *Operations Research, 12*, 300–304.

Barbarosoglu, G., Ozdamar, L., & Cevik, A. (2002). An interactive approach for hierarchical analysis of helicopter logistics in disaster relief operations. *European Journal of Operational Research, 140*(1), 118–133.

Barbarosoglu, G., & Arda, Y. (2004). A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operational Research Society, 55*, 43–53.

Battiti, R., & Tecchiolli, G. (1994). The reactive tabu search. *ORSA Journal on Computing, 6*, 126–140.

Beasley, J. E. (1983). Route first-cluster second methods for vehicle routing. *Omega, 11*, 403–408.

Beasley, J. E. (2002). Population Heuristics. In P. M. Pardalos & M. G. C. Resende (Eds.), *Handbook of Applied Optimization* (pp. 138–156). New York, NY: Oxford University Press.

Ben-Tal, A., Chung, B. D., Mandalab, S. R., & Yao, T. (2011). Robust optimization for emergency logistics planning: Risk mitigation in humanitarian relief supply chains. *Transportation Research Part B: Methodological, 45*(8), 1177–1189.

Berbeglia, G., Cordeau, J-F. , Gribkovskaia, I., & Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *TOP (Trabajos de Investigación Operativa), 15*, 1-31.

Bérubé, J-F., Gendreau, M., & Potvin, J-Y. (2009). A branch-and-cut algorithm for the undirected prize collecting traveling salesman problem. *Networks*, *54*(1), 56–67.

Bianchessi, N., & Righini, G. (2007). Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery. *Computers and Operations Research, 34*(2), 578–594.

Bodin, L. D., Golden, B. L., Assad, A., & Ball, M. O. (1983). Routing and scheduling of vehicles and crews: The state of the art. *Computers and Operations Research, 10,* 63–211.

Brandão, J. (2006). A new tabu search algorithm for the vehicle routing problem with backhauls. *European Journal of the Operational Research*, *173*, 540-555

Brandão, J. (2009). A deterministic tabu search algorithm for the fleet size and mix vehicle routing problem. *European Journal of Operational Research,195*, 716-728 .

Brandeau, M. L. & Chiu, S. S. (1989). An overview of representative problems in location research. *Management Science, 35*(6) 645–674.

Bräysy, O., Gendreau, M., Hasle, G., & Lokketangen, A. (2007a). A survey of heuristics for the vehicle routing problem, Part I: Basic problems and supply side extensions. Working paper, SINTEF ICT, Trondhiem, Norway.

Bräysy, O., Gendreau, M., Hasle, G., & Lokketangen, A. (2007b). A survey of heuristics for the vehicle routing problem, Part II: Demandside extensions. Working paper, SINTEF ICT, Trondhiem, Norway.

Bräysy, O., Dullaert, W., Hasle, G., Mester, D. I., & Gendreau, M. (2008). An effective multirestart deterministic annealing metaheuristic for the fleet size and mix vehicle-routing problem with time windows. *Transportation Science, 42*(3), 371-386.

Breedam, A. V. (1995). Improvement heuristics for the vehicle routing problem based on simulated annealing. *European Journal of Operational Research, 86*, 480-490.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999a). An improved ant system algorithm for the vehicle routing problem. *Annals of Operations Research 89*, 319–328.

Bullnheimer, B., Hartl, R.F., & Strauss, C. (1999b) A new rank-based version of the ant system: A computational study. *Central European Journal of Operations Research Operations Research and Economics. 7*(1), 25–38

Cabot, A., Francis, R., & Strary, M. (1970). A network flow solution to a rectilinear distance facility location problem. *AIIE Transactions, 2*, 132–141.

Caldas, A., Carpente, L., & Lorenzo-Freire, S. (2009). Reactive tabu search algorithm for the pickup and delivery routing problem with time windows. IX Congreso Galego de Estatística e Investigación de Operacións Ourense, 12–13–14 de novembro de 2009.

Cao, E., & Lai, M. (2007). An improved genetic algorithm for the Vehicle Routing Problem with Simultaneous Delivery and Pick-up Service. *Proceedings of the 6th Wuhan International Conference on E-Business* (pp. 2100–2106). Wuhan, China.

Cao, E. (2008, July). A differential evolution and genetic algorithm for vehicle routing problem with simultaneous delivery and pickup. *Proceedings of 17th IFAC World Congress 2008, COEX, Korea, 17*(1), 10576–10581.

Caprara, A., & Fischetti, M. (1997). Branch-and-cut algorithms. In M. Dell'Amico, F. Maffioli, & S. Martello (Eds.), *Annotated Bibliographies in Combinatorial Optimization* (pp. 45–64). New York, NY: Wiley.

Casco, D. O., Golden, B. L., & Wasil, E. A. (1988). Vehicle routing with backhauls: Models, algorithms, and case studies. In B. L. Golden & A. A. Assad (Eds.), *Vehicle Routing: Methods and Studies* (pp. 127–147). Amsterdam: Elsevier.

Çatay, B. (2006). An ant colony optimization approach for the capacitated vehicle routing problem with simultaneous delivery and pick-up. In M. Binetti, F. Civitella, E. De Liddo, M. Dell' Orco, M. Ottomanelli (Eds.), *Proceedings of the 11th Euro Working Group on Transportation (EWGT) Meeting* (pp. 275-282). Bari, Italy.

Çatay, B. (2009). Ant colony optimization and its application to the vehicle routing problem with pickups and deliveries. In R. Chiong & S. Dhakal (Eds.), *Natural Intelligence for Scheduling, Planning and Packing Problems. Studies in Computational Intelligence, 250* (pp. 219–244). Berlin: Springer-Verlag.

Çatay, B. (2010). A new saving-based ant algorithm for the vehicle routing problem with simultaneous pickup and delivery. *Expert Systems with Applications, 37*(10), 6809-6817.

C.D.C., & M.I.A.H. (Centers for Disease Control and Prevention and Merck Institute of Aging and Health). (2004). The state of aging and health in America 2004. CDC, Atlanta, GA and MIAH, Washington, DC.

Chan, L. M. A., Simchi-Levi, D., & Bramel, J. (1995). Worst-case analyses, linear programming and the bin-packing problem. *Mathematical Programming, 83,* 213–227.

Chardaire, P., Lutton, J. L., & Sutter, A. (1995). Thermostatistical persistency: a powerful improving concept for simulated annealing algorithms. *European Journal of Operational Research, 86*, 565–579.

Chen. J. F., & Wu, T. H. (2006). Vehicle routing problem with simultaneous deliveries and pickups. *Journal of the Operational Research Society, 57*, 579-587.

Chiu, Y. C. (2004). Traffic scheduling simulation and assignment for area-wide evacuation. Intelligent Transportation Systems. *Proceedings of the 7th International IEEE Conference on Intelligent Transportation Systems (ITSC 2004)* (pp. 537–542). Washington, DC.

Chiu, Y. C., & Zheng, H. (2007). Real-time mobilization decision for multi-priority emergency response resources and evacuation groups: Model formulation and solution. *Transportation Research: Part E*, *43*, 710–736.

Chiu Y., Zheng H., Villalobos J., & Gautam B. (2007). Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. *IIE Transactions, 39*, 83–94.

Choi, E., Tcha, D.-W. (2007). A column generation approach to the heterogeneous fleet vehicle routing problem. *Computers and Operations Research, 34*(7), 2080–2095.

Christian, P. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research*, *31*, 1985-2002.

Christofides, N., Mingozzi, A., & Toth, P. (1979). The vehicle routing problem. In N. Christofides, A. Mingozzi,, P. Toth,, & C. Sandi (Eds.), *Combinatorial Optimization* (pp. 315–338). Chichester, UK: Wiley.

Christofides, N. (1985). Vehicle routing. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, & D. B. Shmoys (Eds.), *The Traveling Salesman Problem* (pp. 431448). Chichester, UK: Wiley.

Church, R. L., & ReVelle, C. S. (1974). The maximal covering location problem. *Papers in Regional Science, 32*(1), 101–118.

Cioppa, T. M. (2002). *Efficient nearly orthogonal and space-filling designs for high-dimensional complex models*. (Unpublished doctoral dissertation), Naval Postgraduate School, Monterey, CA.

Cioppa, T. M., & Lucas, T. W. (2007). Efficient nearly orthogonal and space-filling Latin hypercubes. *Technometrics, 49*(1), 45–55.

Clarke, G., & Wright, J. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, *12*(4), 568–581.

Congressional Research Service. (2005). *Disaster evacuation and displacement policy: Issues for Congress*, by Keith Bea. CRS Report RS22235, Washington, DC: Government Printing Office.

Cordeau, J-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational Research Society, 52,* 928–936.

Crispim, J., & Brandão, J. (2005). Metaheuristics applied to mixed and simultaneous extensions of vehicle routing problems with backhauls. *Journal of the Operational Research Society*, *56*(11), 1296–1302.

Dantzig, G. B., & Fulkerson, D. R. (1954). Minimizing the number of tankers to meet a fixed schedule. *Naval Research Logistics Quarterly*, *1*, 217–222.

Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management Science, 6*, 80–91.

Dell'Amico, M., Righini, G., & Salani, M. (2006). A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transportation Science, 40*(2), 235–247.

Desrochers, M., & Verhoog, T. W. (1989) *A matching based savings algorithm for the vehicle routing problem*. Technical Report Cahiers du GERAD G-89-04, Ecole des Hautes Etudes Commerciales de Montreal, Canada.

Desrochers, M., Lenstra, J. K., & Savelsbergh, M. W. P. (1990). A classification scheme for vehicle routing and scheduling problems. *European Journal of Operational Research Society, 46,* 322–332.

Desrochers, M., Desrosiers, J., & Solomon, M. M. (1992). A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research, 40*, 342–354.

Deif, I., & Bodin, L. (1984). Extension of the Clarke and Wright algorithm for solving the vehicle routing problem with backhauling. In A. Kidder (Ed.), *Proceedings of the Babson conference on software uses in transportation and logistic management,* (pp. 75–96). Babson Park, FL, USA.

Dethloff, J. (2001). Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up. *OR Spectrum, 23*, 79–96.

Dethloff, J. (2002). Relation between vehicle routing problems: An insertion heuristic for the vehicle routing problem with simultaneous delivery and pick-up applied to the vehicle routing problem with backhauls. *Journal of the Operational Research Society, 53,* 115–118.

Doerner, K., Hartl, R. F., & Reimann, M. (2001). Ants Solve Pickup and Delivery Problems with Full Truck Loads. *Operations Research Proceedings 2000* (pp. 395–400). Berlin: Springer.

Dondo, R., & Cerda, J. (2007). A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European Journal of Operations Research, 176,* 1478–1507.

Dorigo, M. (1992). *Optimization, learning and natural algorithms* (in Italian), PhD dissertation, Dipartimento di Elettronica, Politecnico di Milano, Italy.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996) The ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics-Part B, 26,* 29–41.

Dorigo, M., & Gambardella, L. M. (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Transactions on Evolutionary Computation 1*, 53–66.

Dorigo, M., & Stutzle, T. (2004). *Ant colony optimization*. Cambridge: MIT Press.

Dorigo, M. (2008) Ant colony optimization. *Scholarpedia, 2*(3), 1461.

Dow, K., & Cutter, S. (1998). Crying wolf: Repeat responses to hurricane evacuation orders. *Coastal Management, 26*(4), 237–252.

Drabek, T. E. (1999). Understanding disaster warning responses. *The Social Science Journal, 36*(3), 515–523.

Dreo, J., Petrowski, A., Siarry, P., & Taillard, E. (2003). Metaheuristics for hard optimization. Methods and case studies. Berlin: Springer.

Dueck, G., & Scheurer, T. (1990). Threshold accepting: A general purpose optimization algorithm. *Journal of Computational Physics, 90,* 161–175.

Duhamel, C., Potvin, J-Y., & Rousseau, J-M. (1997). A tabu search heuristic for the vehicle routing problem with backhauls and time windows. *Transportation Science, 31*, 49–59.

Du Merle, O., Villeneuve, D., Desrosiers, J. & Hansen, P. (1999). Stabilized column generation. *Discrete Mathematics, 194,* 229–237.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering, 57*(4), 1472–1483.

Eldar, R. (1992). The needs of elderly persons in natural disasters: Observations and recommendations. *Disasters. 16*(4), 355–358.

Federal Emergency Management Agency. (2004). Disaster Preparedness for people with disabilities. Retrieved from http://www.fema.gov/library/disprepf.shtm

Feo, T. A., & Resende, M. G. C. (1995). Greedy Randomized Adaptive Search Procedures. *Journal of Global Optimization, 6,* 109–133.

Fernandez, L. S., Byard, D., Lin, C. C., Benson, S., & Barbera, J. A. (2002). Frail elderly as disaster victims: emergency management strategies. *Prehospital and Disaster Medicine, 17*(2), 67–74.

Festa, P., & Resende, M. G. C. (2002). GRASP: An annotated bibliography. In C. C. Ribeiro, & P. Hansen (Eds.). *Essays and surveys in metaheuristics* (pp. 325–368). Boston: Kluwer.

Fisher, M. L. (1995). Vehicle routing. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), *Network Routing, Handbooks in Operations Research and Management Science 8* (pp. 1–33). North-Holland, Amsterdam: Elsevier Science Pub Co.

Fitzsimmons, J. (1973). A methodology for emergency ambulance deployment. *Management Science, 19*, 627–636.

Fraser, A. (1957a). Simulation of genetic systems by automatic digital computers 1, Introduction. *Australian Journal of Biological Science, 10*, 484–491.

Fraser, A. (1957b). Simulation of genetic systems by automatic digital computers 2, Effects of linkage on rates of advanced under-selection. *Australian Journal of Biological Science, 10*, 492–499.

Gajpal, Y., & Abad, P. L. (2009a). Multi-ant colony system (MACS) for a vehicle routing problem with backhauls. *European Journal of Operational Research, 196*, 102–117.

Gajpal, Y., & Abad, P. L. (2009b). An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup. *Computers & Operations Research, 36*, 3215–3223.

Galea, E. R. (Ed). (2003). Pedestrian and evacuation dynamics. *Proceedings of the 2nd International Conference on Pedestrian and Evacuation Dynamics, London*. UK: CMS Press, University of Greenwich.

Gambardella, L. M., Taillard, E., & Agazzi, G. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In D. Corne, M. Dorigo, & F. Glover (Eds.), *New ideas in optimization* (pp. 63–76). London: McGraw-Hill.

Gambardella, L. M., Taillard, E. D., & Dorigo, M. (1999). Ant colonies for the quadratic assignment problem. *Journal of the Operational Research Society, 50*, 167–176.

Garey, M. R., & Johnson, D. S. (1978). Strong NP-completeness results: Motivation, examples, and implications. *Journal of the Association for Computing Machinery, 25*(3), 499–508.

Garey, M. R., & Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness,* New York: W. H. Freeman & Co.

Gattermann, P., Waldau, N., & Schreckenberg, M. (Eds.) (2006) Pedestrian and evacuation dynamics. *Proceedings of the 3rd International Conference, Vienna*. Berlin: Springer.

Gencer, C., Top, I., & Aydogan, E. K. (2006). A new intuitional algorithm for solving heterogeneous fixed fleet routing problems: passenger pickup algorithm. *Applied Mathematics and Computation*, *181*, 1552–1567.

Gendreau, M., Laporte, G., Musraganyi, C., & Taillard, E. D. (1999). A tabu search heuristic for the heterogeneous fleet vehicle routing problem. *Computers & Operations Research, 26*(12), 1153–1173.

Gendreaua, M., Laportea, G., & Vigo, D. (1999). Heuristics for the traveling salesman problem with pickup and delivery. *Computers & Operations Research, 26*(7), 699–714.

Gendreau, M., Laporte, G., & Semet, F. (2001). A dynamic model and parallel tabu search heuristic for real-time ambulance relocation. *Parallel Computing, 27*(12), 1641–1653.

Gendreau, M. (2002). Recent advances in tabu search. In C. C. Ribeiro & P. Hansen (Eds.), *Essays and Surveys in Metaheuristics* (pp. 369–378). Boston: Kluwer.

Gendreau, M. (2003). An introduction to tabu search. In F., Glover & G. Kochenberger (Eds.), *Handbook of Metaheuristics* (pp. 37–54). Boston: Kluwer.

Gendreau, M., Potvin, J-Y. (2005). Metaheuristics in combinatorial optimization. *Annals of Operations Research, 140*(1), 189–213.

Gissing, A., Morgan, M., & Ronan, C. (2007). Planning for the inevitable – emergency planning for floods in NSW. *Proceedings of 47th NSW Floodplain Management Authorities Conference*.

Gillett, B. E., & Miller, L. R. (1974). A heuristic algorithm for the vehicle dispatch problem. *Operations Research, 22*, 340–349.

Glover, F. (1989). Tabu search, Part I. *ORSA Journal of Computing, 1*, 190–206.

Glover, F. (1990). Tabu search, Part II. *ORSA Journal of Computing, 2*, 4–32.

Glover, F., & Laguna, M. (1997). *Tabu search*. Boston: Kluwer.

Glover, F., & Kochenberger, G. (2003). *Handbook of metaheuristics.* Dordrecht: Kluwer Academic Publishers.

Goetschalckx, M, & Jacobs-Blecha, C. (1989). The vehicle routing problem with backhauls, *European Journal of Operational Research, 42(*1), 39–51.

Goetschalckx, M., & Jacobs-Blecha, C. (1993). *The vehicle routing problem with backhauls: Properties and solution algorithms.* Technical Report MHRC-TR.88-13, Georgia Institute of Technology.

Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. (1977). Implementing vehicle routing algorithms. *Networks, 7*, 113–148.

Golden, B. L., Baker, E. K., Alfaro, J. L., & Schaffer, J. R. (1985). The vehicle routing problem with backhauling: two approaches, Working Paper Series MS/S 85–017, University of Maryland, College Park.

Golden, B. L., & Skiscim, C. C. (1986). Using simulated annealing to solve routing and location problems. *Naval Research Logistics Quarterly 33*, 261–279.

Golden, B. L., & Assad, A. A. (1988). *Vehicle Routing: Methods and Studies.* Amsterdam: North-Holland.

Golden, B. L., Wasil, E. A., Kelly., J. P., & Chao, I. M. (1998). Metaheuristics in vehicle routing. In T. G. Crainic, & G. Laporte (Eds.), *Fleet Management and Logistics* (pp. 33–56), Boston, MA: Kluwer.

Guha-Sapir, D., Vos, F, Below, R., & Ponserre, S. (2011) *Annual Disaster Statistical Review 2010: The Numbers and Trends*. Brussels: Centre for Research on the Epidemiology of Disasters (CRED).

Hadjiconstantinou, E., Christofides, N., & Mingozzi, A. (1995). A new exact algorithm for the vehicle routing problem based on q-paths and ^-shortest paths relaxations. *Annals of Operations Research, 61*, 21–43.

Haghani, A., & Oh, S. C. (1996). Formulation and solution of a multi-commodity, multi-modal network flow model for disaster relief operations. *Transportation Research, 30*(3), 231–250.

Haimovich, M., & Rinnooy Kan, A. H. G. (1985). Bounds and heuristics for capacitated routing problems. *Mathematics of Operations Research, 10*, 527–542.

Halse, K. (1992). *Modeling and solving complex vehicle routing problems (*Doctoral dissertation). Institute of Mathematical Statistics and Operations Research (IMSOR), Technical University of Denmark, Lyngby.

Hasle, G., & Kloster, O. (2007). *Industrial vehicle routing*. In *Geometric modeling, numerical simulation, and optimization: applied mathematics at SINTEF* (pp. 397–435). Berlin: Springer.

Heath, S. K., Bard, J. F., & Morrice, D. J. (2012). A GRASP for simultaneously assigning and sequencing product families on flexible assembly lines. *Annals of Operations Research* (13 June 2012), 1–29, doi:10.1007/s10479-012-1167-5

Henderson, D., S. H. Jacobson, & A. W. Johnson. (2003). The theory and practice of simulated annealing. In F. Glover, & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 287–319). Boston: Kluwer.

Ho, W., Ho, G. T. S., Ji, P., & Lau, H. C. W. (2008). A hybrid genetic algorithm for the multi-depot vehicle routing problem. *Engineering Applications of Artificial Intelligence*, *21*(4), 548–557

Holland, J. H. (1975). *Adaptation in natural and artificial systems*. Ann Arbor, MI: University of Michigan Press.

Hosny, M. I. (2010). *Investigating heuristic and meta-heuristic algorithms for solving pickup and delivery problems* (Doctoral dissertation). Cardiff University, School of Computer Science & Informatics. Retrieved from http://faculty.ksu.edu.sa/M_F_Hosny/Documents/Thesis.pdf

Houle, J. M. (2011). *Many-to-many vehicle routing with heterogeneous vehicle and passenger classes* (Master's thesis). Graduate School of the University of Minnesota.

Institute of Food and Agricultural Sciences. (1999). Disaster planning for elderly and disabled populations. Retrieved from http://disaster.ifas.ufl.edu/PDFS/CHAP02/D02-09.PDF

Joubert, J. W., & Claasen, S. J. (2006). A sequential insertion heuristic for the initial solution to a constrained vehicle routing problem. *Orion, 22*(1), 105–116.

Jünger, M. , Reinelt, G., & Rinaldi, G. (1995). The traveling salesman problem. In M. O. Ball, T. L. Magnanti, C. L. Monma, & G. L. Nemhauser (Eds.), *Network Models, Handbooks in Operations Research and Management Science 7* (pp. 225–330). Amsterdam: North-Holland.

Jünger, M., & Thienel, S. (1998). Introduction to ABACUS – A branch-and-cut system. *Operations Research Letters, 22,* 83–95.

Kanthavel, K., & Prasad, P. (2010). Optimization of capacitated vehicle routing problem by nested particle swarm optimization. *American Journal of Applied Sciences, 8*(2), 107–112.

Kim, S., & Shekhar, S. (2005). Contraflow network reconfiguration for evacuation planning: a summary of results. *Proceedings of 13th Annual ACM International Workshop on Geographic Information Systems (pp.* 250–259).

Kirckpatrick, S., Gelatt, Jr., C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220,* 671–680.

Kleijnen, J. P. C., Sanchez, S. M., Lucas, T. W., & Cioppa, T. M. (2005). A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal on Computing, 17*(3), 263–289.

Klose, A., & Drexl, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research, 162*(1), 4–29.

Kovacs, G., & Spens, K. M. (2007). Humanitarian logistics in disaster relief operations. *International Journal of Physical Distribution & Logistics Management, 37*(2), 99–114.

Kuba, M., Dorian, A., Kuljian, S., & Shoaf, K. (2004). *Elderly populations in disasters: recounting evacuation processes from two skilled-care facilities in central Florida, August 2004.* Quick Response Report. 172. Natural Hazards Center and Applications Information, University of Colorado, Boulder, CO.

Lämmel, G., Rieser, M., & Nagel, K. (2008). Bottlenecks and Congestion in Evacuation Scenarios: A Microscopic Evacuation Simulation for Large-Scale Disasters. *Proceedings of 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*. Estoril, Portugal.

Laporte, G., & Nobert, Y. (1987). Exact algorithms for the vehicle routing problem. *Annals of Discrete Mathematics, 31*, 147–184.

Land, M. (1998). *Evolutionary algorithms with local search for combinatorial optimization* (Doctoral dissertation). University of California, San Diego. Retrieved from http://scholar.google.com/scholar?q=Land+%2BEvolutionary+algorithms+with+local+search+for+combinatorial+optimization&hl=en&as_sdt=0%2C5&as_vis=1

Laporte, G. (1992). The vehicle routing problem: an overview of exact and approximate algorithms. *European Journal of Operational Research, 59*, 345–358.

Laporte, G., & Osman I. H. (1995) Routing problems: A bibliography. *Annals of Operations Research, 61*, 227–262.

Laporte, G., & Osman I. H. (1996) Metaheuristics in combinatorial optimization. *Annals of Operations Research, 63*, 57–75.

Laporte, G. (1997). Vehicle routing. In M. Dell'Amico, F. Maffioli, & S. Martello (Eds.), *Annotated Bibliographies in Combinatorial Optimization*, Chichester, UK : Wiley.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation Science, 43*(4), 408–416.

Lau, H. C. W., Chan, T. M., Tsui, W. T., & Pang, W. K. (2010). Application of genetic algorithms to solve the multidepot vehicle routing problem. *IEEE Transactions on Automation Science and Engineering, 7*(2), 383–392.

Le Louarn, F-X., Gendreau, M., & Potvin, J-Y. (2004). GENI ants for the traveling salesman problem. *Annals of Operations Research, 131*, 187–201.

Lenstra, J. K., Rinnooy Kan, A. H. G. (1981). Complexity of vehicle routing and scheduling problems. *Networks*, 11, 221–227

Li, F., Golden, B., & Wasil, E. (2007). A record-to-record travel algorithm for solving the heterogeneous fleet vehicle routing problem. *Computers and Operations Research, 34*, 2734–2742.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell System Technical Journal, 44*, 2245–2269.

Liu, S., Huang, W., & Ma, H. (2009). An effective genetic algorithm for the fleet size and mix vehicle routing problems. *Transportation Research Part E, 45*(2009), 434–445.

Lu, Q., Huang, Y., & Shekhar, S. (2003). Evacuation planning: a capacity constrained routing approach. In H. Chen, R. Miranda, D. D. Zeng, C. C. Demchak, J. Schroeder, T. Madhusudan (Eds.), *ISI 2003. LNCS*, *2665*, 111–125. Heidelberg: Springer.

Lu, Q., & Dessouky, M. M. (2004). An exact algorithm for the multiple vehicle pickup and delivery problem. *Transportation Science, 38*, 503–514.

Lu, Q. (2005). A new insertion-based construction heuristic for solving the pickup and delivery problem with hard time windows. *European Journal of Operational Research, 175*, 672–687.

Lu, Q., George, B., Shekhar, S. (2005). Capacity Constrained Routing Algorithms for Evacuation Planning: A Summary of Results. In E. Anshelevich, M. J. Egenhofer, & J. Hwang (Eds.), *SSTD 2005. LNCS*, 3633, 291–307. Heidelberg: Springer.

Lu, Q. (2006). *Capacity constrained routing algorithms for evacuation route planning* (Doctoral dissertation). University of Minnesota.

MacLellan, J. I., & Martell, D. (1996). Basing airtankers for forest fire control in Ontario. *Operations Research, 44*(5), 677–686.

Magnanti, T. L. (1981). Combinatorial optimization and vehicle fleet planning: Perspectives and prospects. *Networks, 11*, 179–214.

Mamada, S., Makino, K., & Fujishige, S. (2004). Evacuation problems and dynamic network flows. SICE Annual Conference in Sapporo, Hokkaido Institute of Technology, Japan, 530–53.

Marinakis, Y. & Migdalas, A. (2007). Annotated bibliography in vehicle routing. *Operational Research, 7*(1), 27–46.

McGuire, L. C., Ford, E. S. & Ajani, U. A. (2006). Cognitive functioning as a predictor of functional disability in later life. *American Journal of Geriatric Psychiatry. 14*(1), 36–42.

McGuire, L., Ford, E., & Okoro, C. (2007). Natural disasters and older U.S. adults with disabilities: Implications for evacuation. *Disasters*, *31*(1), 49−56.

Michalewicz, Z. (1996). *Genetic Algorithms+Data Structures=Evolution Programs*. Third Edition. Berlin: Springer.

Min, H. (1989). The multiple vehicle routing problem with simultaneous delivery and pick-up points. *Transportation Research A, 23A,* 377–386.

Min, H., Current, J., & Schilling, D. (1992). The multiple depot vehicle routing problem with backhauling. *Journal Of Business Logistics, 13*, 259–288.

Mingozzi A, Giorgi S., & Baldacci R. (1999). An exact method for the vehicle routing problem with backhauls. *Transportation Science, 33*, 315–329.

Mokdad, A.H. Mensah, G.A., Posner, S. F., Reed, E., Simoes, E. J., Engelgau, M. M., & the Chronic Diseases and Vulnerable Populations in Natural Disasters Working Group (2005). When chronic conditions become acute: prevention and control of chronic diseases and adverse health outcomes during natural disasters. *Preventing Chronic Disease. 2 (Special Issue)*. November. Retrieved from: http://www.cdc.gov/PCD/issues/2005/nov/05_0201.htm

Morrow, B. H. (1999). Identifying and mapping community vulnerability. *Disasters. 23*(1), 1–18.

Moscato, P. (2002). Memetic Algorithms. In P.M. Pardalos, & M.G.C. Resende (Eds.), Handbook of Applied Optimization (pp. 157–167). New York: Oxford University Press.

Mosheiov, G. (1994). The traveling salesman problem with pick-up and delivery. *European Journal of Operational Research, 79,* 299–310.

Mosheiov, G. (1998). Vehicle routing with pickup and delivery: Tour-partitioning heuristics. *Computers and Industrial Engineering, 34*, 669–684.

Nagy, G. (1996). *Heuristic methods for the many-to-many location-routing problem* (Doctoral dissertation). School of Mathematics and Statistics, University of Birmingham, Birmingham.

Nagy, G., & Salhi, S. (1998). The many-to-many location-routing problem. *TOP (Trabajos de Investigación Operativa), 6,* 261–275.

Nagy, G., & Salhi, S. (2005). Heuristic algorithms for single and multiple depot vehicle routing problems with pickups and deliveries. *European Journal of Operational Research, 162*(1), 126–141.

Nagy, G., & Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research, 177*, 649–672.

Nelson, M. D., Nygard, K. E., Griffin, J. H., & Shreve, W. E. (1985). Implementation techniques for the vehicle routing problem. *Computers and Operations Research, 12*, 273–283.

Nemhauser, G. L., & Wolsey, L. A. (1988). *Integer and Combinatorial Optimization*. Chichester, UK: Wiley.

Nieburg, P., Waldman, R. J., & Krumm., D. M. (2005). Evacuated populations - Lessons from foreign refugee crises. *The New England Journal of Medicine, 353*(15), 1547–1549.

Nisha de Silva, F. (2001). Providing special decision support for evacuation planning: a challenge in integrating technologies. *Disaster Prevention and Management, 10*(1), 11–20.

Ochi, L. S., Vianna, D. S., Drummond, L. M., & Victor, A. O. (1998). A parallel evolutionary algorithm for the vehicle routing problem with heterogeneous fleet. *Future Generation Computer System*, *14*, 285–292.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research, 41*, 421–451.

Osman, I. H. (1995) An introduction to metaheuristics. In Lawrence, M., & Wilsdon C. (Eds), *Operational Research Tutorial Papers*. Birmingham: Operational Research Society.

Osman I., & Kelly, J. (1996) Metaheuristics. An overview. In I. H. Osman & J. P. Kelly (Eds.), *Metaheuristics, theory and applications*. Boston: Kluwer.

Osman, I. H., & Laporte, G. (1996) Metaheuristics: A bibliography. *Annals of Operations Research on Metaheuristics 63*. Baltzer, Basel.

Osman, I. H., & Wassan, N. A. (2002). A reactive tabu search meta-heuristic for the vehicle routing problem with backhauls. *Journal of Scheduling, 5*, 263–285.

Osman, I., & Salhi, S., (1996). Local search strategies for the vehicle fleet mix problem. In V. J. Rayward-Smith, I. H. Osman, C. R. Reeves, & G. D. Smith (Eds.), *Modern Heuristic Search Methods* (pp. 131–153). New York: Wiley.

Osman, M. (2010). *Evacuation route scheduling approaches on transport networks that combine building pathways and roads* (doctoral dissertation). North Carolina A&T State University.

Osman, M.F., & Ram, B. (2011). Evacuation route scheduling using discrete time-based capacity-constrained model. IEEE International Conference Industrial Engineering and Engineering Management (IEEM). 6–9 Dec 2011, 161 – 165.

Ozdamar, L., Ekinci, E., & Kucukyazici, B. (2004). Emergency logistics planning in natural disaster. *Annals of Operations Research, 129*, 217–245.

Özdamar L., & Yi, W. (2008). Greedy neighborhood search for disaster relief and evacuation logistics. *IEEE Intelligent Systems, 23*(1), 14–23.

Padberg, M., & Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large scale symmetric traveling salesman problems. *SIAM Review, 33*, 60–100.

Palmer, K., Dessouky, M. M., & Abdelmaguid, T. (2004). Impacts of management practices and advanced technologies on demand responsive transit systems. *Transportation Research, Part A; Policy and Practice, 38,* 495–509.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008a). A survey on pickup and delivery problems. Part I: Transportation between customers and depot. *Journal für Betriebswirtschaft, 58,* 21–51.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008b). A survey on pickup and delivery problems. Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft, 58,* 81–117.

Perry, R. (1985). *Comprehensive emergency management: evacuating threatened populations*. London: Jai Press.

Potvin, J. Y. (1996). Genetic algorithms for the traveling salesman problem. *Annals of Operations Research, 63*, 339–370.

Potvin J. Y., Duhamel C., & Guertin, F. (1996). A genetic algorithm for vehicle routing with backhauling. *Applied Intelligence, 6*, 345–355.

Prais, M., & Ribeiro, C. C. (2000). Parameter Variation in GRASP procedures. *Investigacion Operativa, 9,* 1–20.

Prins, C. (2004). A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers & Operations Research, 31*(2), 1985–2002.

Rathi, A. K., Church, R. L., & Solanki, R. S. (1992). Allocating resources to support a multicommodity flow with time windows. *Logistics and Transportation Review, 28*(2), 167–188.

Reeves, C. R. (Ed). (1993). *Modern heuristic techniques for combinatorial problems*. New York, NY: John Wiley & Sons, Inc.

Reimann, M., Doerner, K., & Hartl, R.F. (2004). D-Ants: Savings based ants divide and conquer the vehicle routing problem. *Computers and Operations Research, 31*, 563–591.

Resende, M. G. C., & Ribeiro, C. C.. (2003). Greedy randomized adaptive search procedures. In F. Glover & G. Kochenberger (Eds.), *Handbook of metaheuristics* (pp. 219–249). Boston: Kluwer.

ReVelle, C., Bigman, D., Schilling, D., Cohon, J., & Church, R. (1977). Facility location: A review of context-free and EMS models. *Health Services Research, 12*(2), 129–146.

Ribeiro, C., & Soumis, F. (1994). A column generation approach to the multi-depot vehicle scheduling problem. *Operations Research, 42*, 41–52.

Rizzoli, A. E., Oliverio, F., Montemanni, R., & Gambardella, L. M., (2004). Ant Colony Optimisation for vehicle routing problems: from theory to applications. *Galleria Rassegna Bimestrale Di Cultura, 9*(1), 1–50.

Robin, D. W. (2000). Preparing long-term care facilities for natural disasters. *Annals of Long-term Care*. Retrieved from: http://www.annalsoflongtermcare.com/article/950

Ropke, S., & Pisinger, D. (2006). A unified heuristic for a large class of vehicle routing problems with backhaul. *European Journal of Operational Research, 171,* 750–775.

Salhi, S., & Rand, G. K. (1993). Incorporating vehicle routing into the vehicle fleet composition problem. *European Journal of Operational Research, 66,* 313–330.

Salhi, S., & Sari, M. (1997). Models for the multi-depot vehicle fleet mix problem. *European Journal of Operational Research, 103*, 95–112.

Salhi, S., & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society, 50*, 1034–1042.

Saliba, D., Buchanan, J., & Kington, R. S. (2004). Function and response of nursing facilities during community disaster. *American Journal of Public Health, 94*(8), 1436–1441.

Salmeron J., & Apte, A. (2010) Stochastic optimization for natural disaster asset prepositioning, *Production and Operations Management, 19*(5), 561–574.

Sanchez, S. M. (2005). NOLH designs spreadsheet. Retrieved from http://harvest.nps.edu

Savelsbergh, M. W. P., & Sol, M. (1995). The general pickup and delivery problem. *Transportation Science, 29*, 17–29.

Schreckenberg, M., & Sharma, S. D. (Eds). (2001). Pedestrian and evacuation dynamics. *Proceedings of the 1st International Conference, Duisburg*.

Sherali, H. D., Carter, T. B., & Hobeika, A. G. (1991). A location-allocation model and algorithm for evacuation planning under hurricane/flood conditions. *Transportation Research Part B: Methodological, 25*(6), 439–452.

Shekhar, S., & Kim, S. (2006). *Contraflow transportation network reconfiguration for evacuation route planning.* Technical Report, Mn/DOT 2006-21, Department of Computer Science and Engineering, University of Minnesota.

Sheu, J. B. (2007). An emergency logistics distribution approach for quick response to urgent relief demand in disasters. *Transportation Research: Part E*, *43*, 687–709.

Shin, K. (2009). A geometrical center based two-way search heuristic algorithm for vehicle routing problem with pickups and deliveries. *Journal of Information Processing Systems, 5*(4), 237–242.

Shmoys, D. B., Tardos, E., & Aardal, K. I.(1997). Approximation algorithms for facility location problems. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (pp. 265–274).

Shtovba, S. D. (2005). Ant algorithms: Theory and applications, programming and computing software (Translated from *Programmirovanie*), *31*(4), 167–178.

Simonovic, S. P., & Ahmad, S. (2005). Computer-based model for flood evacuation emergency planning. *Natural Hazards, 34*(1), 25–51.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time windows, *Operations Research, 35*(2), 254–265.

Solomon, M. M., & Desrosiers, J. (1988). Survey paper – Time window constrained routing and scheduling problems. *Transportation Science, 22*(1), 1–13.

Sombuntham, P., & Kachitvichayanukul, V. (2010). A Particle Swarm Optimization Algorithm for Multi-depot Vehicle Routing problem with Pickup and Delivery Requests. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010* (pp. 1998–2003).

Sorensen, J. H., Carnes, S. A. & Rogers, G. O. (1992). An approach for deriving emergency planning zones for chemical munitions emergencies, *Journal of Hazardous Materials, 30*(3), 223–242.

Stepanov A., & Smith J. M. (2008). Multi-objective evacuation routing in transportation networks. *European Journal of Operational Research, 198,* 435–446.

Stutzle, T., & Hoos, H. H. (1997). The MAX-MIN ant system and local search for the travelling salesman problem. In T. Black, Z. Michalewicz, & X. Yao (Eds.), *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation (ICEC 1997)* (pp. 309–314). Piscataway: IEEE Press.

Stutzle, T., & Hoos, H. H. (2000). MAX-MIN Ant System. *Future Generation Computer Systerms*. *16*(8), 889–914.

Taillard, E. (1999). A heuristic column generation method for the heterogeneous fleet VRP. *Operations Research, 33*(1), 1–14.

Tang, F. A., & Galvão, R. D. (2002). Vehicle routing problems with simultaneous pick-up and delivery service. *Journal of the Operational Research Society of India (OPSEARCH), 39*, 19–33.

Tang, F. A., & Galvão, R. D. (2006). A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Computers & Operations Research, 33*, 595–619.

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2003). A list based threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *Journal of the Operational Research Society, 54*, 65–71.

Tarantilis, C. D., Kiranoudis, C. T., & Vassiliadis, V. S. (2004). A threshold accepting metaheuristic for the heterogeneous fixed fleet vehicle routing problem. *European Journal of Operational Research, 152*, 148–158.

Teng, W. H., Hsu, M. H., Wu, C. H., Chen, & A. S. (2006). Impact of flood disasters on Taiwan in the last quarter century. *Natural Hazards, 37*(1), 191–207.

Thangiah, S.R., Potvin, J-Y., & Sun, T. (1996). Heuristic approaches to vehicle routing with backhauls and time windows. *Computers & Operations Research, 23,* 1043–1057.

Thienel, S. (1995). *ABACUS: A Branch and Cut System* (Doctoral dissertation). Universitat zu Koln, Germany.

Thomas, A. (2003). Humanitarian logistics: Enabling disaster response, Fritz Institute, pp. 15.

Thomas, A. (2005). Matching recognition with responsibility. *Freight and Logistics*. Retrieved from: http://www.fritzinstitute.org/PDFs/InTheNews/2005/ADR 0605.pdf

Thomas, A., & Kopczak, L. (2005). From logistics to supply chain management. The path forward in the humanitarian sector. Fritz Institute, Retrieved from: www.fritzinstitute.org/PDFs/WhitePaper/FromLogisticsto.pdf

Thomas, A., & Mizushima, M. (2005). Logistics training: Necessity or luxury? *Forced Migration Review*, *22*, 60–61.

Thompson, P. M., & Psaraftis, H. N. (1993). Cyclic transfer algorithms for multi-vehicle routing and scheduling problems. *Operations Research, 41,* 935–946.

Toregas, C. R., Swain, R., ReVelle, C. S., & Bergman, L. (1971). The location of emergency service facilities. *Operations Research*, *19*(6), 1363–1373.

Toth, P., & Vigo, D. (1997). An exact algorithm for the vehicle routing problem with backhauls. *Transportation Science, 31*, 372–385.

Toth, P., & Vigo, D. (1998). Exact algorithms for vehicle routing. In T. Crainic and G. Laporte (Eds.), *Fleet management and logistics* (pp. 1–31). Boston, MA: Kluwer.

Toth, P., & Vigo, D. (1999). A heuristic algorithm for the symmetric and asymmetric vehicle routing problems with backhauls. *European Journal of Operation Research*, *113,* 528–543.

Toth, P., & Vigo, D. (Eds.). (2002). *The vehicle routing problem*. Philadelphia, PA: Society for Industrial and Applied Mathematics.

Trunick, P. A. (2005). Logistics when it counts. *Logistics Today, 46*(2), p. 38.

U.S. Census Bureau. (2001). Americans with Disabilities 1997. Washington, DC: U.S. Census Bureau.

Van Breedam, A. (1994). *An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints* (Doctoral dissertation). University of Antwerp.

Van Wassenhove, L. N. (2006). Humanitarian aid logistics: Supply chain management in high gear. *Journal of Operational Research Society, 57*(5), 475–489.

Vogt, B., & Sorensen, J. (1992). *Evacuation research: A reassessment*. Technical report, Oak Ridge National Lab., Tennessee, United States.

Wade, A. C., & Salhi, S. (2002). An investigation into a new class of vehicle routing problem with backhauls. *Omega, 30* (6), 479–487.

Wade, A. C., & Salhi, S. (2003). An ant system algorithm for the mixed vehicle routing problem with backhauls. In M. G. Resende, & J. P. de Sousa (Eds.), *Metaheuristics: Computer Decision-Making* (pp. 699–719). New York: Kluwer.

Wang, X., Golden, B., & Wasil, E. (2008). Using a genetic algorithm to solve the generalized orienteering problem. In B. Golden, S. Raghavan, & E. Wasil (Eds.), *The Vehicle Routing Problem: Latest advances and new challenges* (pp. 263–274). Berlin: Springer.

Wassan, N. A., & Osman, I. H. (2002). Tabu search variants for the mix fleet vehicle routing problem. *Journal of the Operational Research Society,* 53(7), 768–782.

Wassan, N. A. (2007). Reactive tabu adaptive memory programming search for the vehicle routing problem with backhauls. *Journal of the Operational Research Society, 58*, 1630–1641.

Wassan, N. A., Nagy, G., & Ahmadi, S. (2008). A heuristic method for the vehicle routing problem with mixed deliveries and pickups. *Journal of Scheduling, 11*, 149–161.

Yaman, H. (2006). Formulations and valid inequalities for the heterogeneous vehicle routing problem. *Mathematical Programming, 106*(2), 365–390.

Yellow, P. (1970). A computational modification to the savings method of vehicle scheduling. *Operational Research Quarterly, 21*, 281–283.

Yi, W., & Kumar, A., (2007). Ant colony optimization for disaster relief operations. *Transportation Research Part E, 43*, 660–672.

Yi-Chang, C., Hong, Z., Jorge, V., & Bikash, G., (2007). Modeling no-notice mass evacuation using a dynamic traffic flow optimization model. *IIE Transactions, 39*(1), 83–94.

Yuan, Y., & Wang, D. (2009). Path selection model and algorithm for emergency logistics management. *Computers & Industrial Engineering, 56*, 1081–1094.

Yusoff, M., Ariffin, J., & Mohammed, A. (2011). A multi-valued discrete particle swarm optimization for the evacuation vehicle routing problem. *Proceedings of the Second International Conference on Advances in Swarm Intelligence ICSI'11, Volume Part I* (pp. 182–193).

Zhao, F., Li, S., Sun, J., & Mei, D. (2008) Genetic algorithm for the one-commodity pickup-and-delivery traveling salesman problem. *Computers & Industrial Engineering*, *56*(4), 1642–1648.

Zhao, F., Mei, D., Sun, J., & Liu, W. (2009, June). A hybrid genetic algorithm for the vehicle routing problem with simultaneous pickup and delivery, *Proceedings from the 21st Annual International Conference on Chinese Control and Decision Conference* (pp. 3964–3969). Guilin, China.

Zong, X., Xiong, S., Fang, Z., & Li, Q. (2010, July). Multi-ant colony system for evacuation routing problem with mixed traffic flow. *Proceedings of Congress on Evolutionary Computation* (pp. 1–6). Barcelona, Spain.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Dr. Susan Heath
   Naval Postgraduate School
   Monterey, California

4. Dr. Aruna Apte
   Naval Postgraduate School
   Monterey, California